CNC 8070







(REF. 0509)

(SOFT V03.0X)

编程手册

(Soft V03.0x)

(Ref. 0509)

目 录

第 1 章 程序生成

1.1 程序结构 1
1.2 程序段结构 4
1.3 ISO 代码编程 5
1.3.1 准备 "G" 功能表 7
1.4 高级语言编程 10
1.5 参数、常数和表达式 11
1.5.1 算数参数 12
1.5.2 操作和功能 14
1.5.3 表达式 17

第 2 章 机床概述

2.1 轴的命名 19
2.2 坐标系统 21
2.3 参考系统 22
2.3.1 参考系原点 23
2.4 回原点搜索 24
2.4.1 定义 " 回原点搜索 " 24
2.4.2 " 回原点搜索 " 编程 25

第 3 章 坐标系统

3.1 平面选择 (G17/G18/G19/G20) 27
3.1.1 编写两方向定义工作平面 (G20) 29
3.1.2 垂直向刀具轴选择 30
3.2 公制 (G71) 和英制 (G70) 编程 31
3.3 绝对坐标 (G90) 和增量坐标 (G91) 32
3.4 半径编程 (G152) 和直径编程 (G151) 33
3.5 编程坐标 34
3.5.1 笛卡尔坐标 34
3.5.2 极坐标 35

第 4 章 原点选择

4.1 相对于机床零点编程 39
4.2 夹具偏置 41
4.3 坐标预制 (G92) 43
4.4 零点偏置 (G54-G59/G159) 44
4.4.1 增量零点偏置 (G158) 46
4.4.2 零点偏置排除的轴 (G157) 48
4.5 零点偏置取消 (G53) 49
4.6 极坐标原点预置 (G30) 50

第 5 章 工艺功能

5.1 加工进给率 (F) 51
5.2 进给率相关功能 53
5.2.1 进给率单位编程 (G93/G94/G95) 53
5.2.2 进给率混合 (G108/G109/G193) 55
5.2.3 恒进给率方式 (G197/G196) 57
5.2.4 进给率修调的 % 的取消 (G266) 59
5.2.5 加速度控制 (G130/G131) 60
5.2.6 加加速度控制 (G132/G133) 62
5.2.7 向前反馈控制 (G134) 63
5.2.8 AC- 向前控制 (G135) 64
5.3 主轴速度 (S) 65
5.3.1 主轴速度编程 66
5.3.2 钻速极限 67
5.4 刀具号码 (T) 68
5.5 刀具偏置号码 (D) 71
5.6 辅助 (混杂的) 功能 (M) 73
5.6.1 "M" 功能表 74
5.7 辅助功能 (H) 79

第 6 章 刀具路径控制

6.1 快速定位 (G00) 81
6.2 直线插补 (G01) 83
6.3 圆弧插补 (G02/G03) 86
6.3.1 笛卡尔坐标 (圆弧中心编程) 88
6.3.2 笛卡尔坐标 (半径编程) 89
6.3.3 极坐标 92
6.3.4 临时极坐标原点移位至圆弧中心 (G31) 95
6.3.5 绝对坐标圆心 (G06/G261/G262) 96
6.3.6 圆心中心修正 (G264/G265) 97
6.4 与前一路径相切圆弧 (G08) 98
6.5 三点定义圆弧 (G09) 99
6.6 螺旋插补 (G02/G03) 100
6.7 恒螺距螺纹加工 (G33) 102
6.8 刚性攻螺纹 (G63) 107
6.9 手动干涉 (G200/G201/G202) 110
6.9.1 附加手动干涉 (G201/G202) 111
6.9.2 专用手动干涉 (G200) 112

第 7 章 几何协助

7.1 方角 (G07/G60) 113
7.2 半圆角 (G50) 114
7.3 可控圆角，半径混合 (G05/G61) 115
7.3.1 圆角类型 116
7.4 圆角 , 半径混合 (G36) 120
7.5 拐角斜面 (G39) 122
7.6 切入 (G37) 124
7.7 切出 (G38) 125
7.8 镜像 (G11, G12, G13, G10, G14) 126
7.9 坐标系旋转 , 图形旋转 (G73) 129
7.10 缩放因子 131

第 8 章 附加准备功能

8.1 暂停 (G04) 133
8.2 软行程极限编程 (G198-G199) 134
8.3 Hirth 轴 (G170-G171) 135
8.4 OEM 子程序 (G180-G189) 136
8.5 轴的参数范围的变化 (G112) 138
8.6 探测 (G100) 139
8.6.1 包含 / 排斥探针偏置 (G101/G102) 140

第 9 章 刀具补偿

9.1 刀具半径补偿 144
9.1.1 相关刀具半径补偿功能 145
9.1.2 刀具半径补偿建立 148
9.1.3 刀具半径补偿选择 152
9.1.4 加工时刀具半径补偿变换 156
9.1.5 刀具半径补偿取消 158
9.2 刀具长度补偿 161

第 10 章 铣削固定循环

10.1 概述 163
10.1.1 固定循环定义 165
10.1.2 固定循环的影响范围 165
10.1.3 固定循环取消 165
10.1.4 工作平面 166
10.1.5 编程命令 167
10.1.6 在其它平面编程 168
10.2 G81. 钻削固定循环 170
10.2.1 编程实例 171
10.3 G82. 可变啄式深孔钻削循环 172
10.3.1 编程实例 175
10.4 G83. 不变啄式深孔钻削循环 176
10.4.1 编程实例 178
10.5 G84. 攻丝固定循环 179
10.5.1 编程实例 181
10.6 G85. 铰孔固定循环 182
10.6.1 编程实例 183

10.7 G86. 镗孔固定循环 184
10.7.1 编程实例 185
10.8 G87. 矩形腔加工固定循环 186
10.8.1 编程实例 189
10.9 G88. 圆柱型腔加工固定循环 191
10.9.1 编程实例 195

第 **11** 章 **多重加工**

11.1 G160. 在直线模式上的多重加工 198
11.1.1 编程实例 200
11.2 G161. 在平行四边形模式上的多重加工 201
11.2.1 编程实例 203
11.3 G162. 在网格模式上的多重加工 204
11.3.1 编程实例 206
11.4 G163. 在整圆模式上的多重加工 207
11.4.1 编程实例 209
11.5 G164. 在圆弧模式上的多重加工 210
11.5.1 编程实例 212
11.6 G165. 在弦模式上的多重加工 213
11.6.1 编程实例 215

第 **12** 章 **循环编辑**

12.1 概述 217
12.1.1 联合固定循环的多重加工操作 219
12.1.2 加工运动 221
12.1.3 数据、轮廓、图表选择 222
12.1.4 当参数值为 0 时应用的数值 223
12.1.5 模拟固定循环 224
12.2 中心钻 226
12.2.1 基本操作 : 227
12.3 钻削 1. 228
12.3.1 基本操作 : 229
12.4 钻削 2. 230
12.4.1 基本操作 : 231
12.5 攻螺纹 232
12.5.1 基本操作 : 233
12.6 铰削 234
12.6.1 基本操作 : 235
12.7 镗削 1. 236
12.7.1 基本操作 : 237
12.8 镗削 2. 238
12.8.1 基本操作 : 239
12.9 简单行腔加工 240
12.9.1 基本操作 : 242
12.10 矩形行腔加工 243
12.10.1 基本操作 : 247
12.11 圆形行腔加工 248
12.11.1 基本操作 : 252
12.12 预制空行腔加工 253
12.12.1 基本操作 : 256
12.13 2D 行腔加工 258
12.13.1 可执行行腔加工文件 262
12.13.2 基本操作 263
12.13.3 如何定义 2D 轮廓的实例 265
12.14 3D 行腔加工 268
12.14.1 可执行行腔加工文件 273
12.14.2 基本操作 274
12.14.3 如何定义 3D 轮廓的实例 275
12.15 矩形凸台加工 279
12.15.1 基本操作 : 282
12.16 圆形凸台加工 283
12.16.1 基本操作 286
12.17 表面铣削 287
12.17.1 基本操作 290
12.18 点到点轮廓 i 291
12.18.1 基本操作 294
12.19 轮廓 295
12.19.1 基本操作 298
12.20 窄槽铣削 299
12.20.1 基本操作 302
12.21 在直线模式上的多重加工 304
12.22 在圆弧模式上的多重加工 305
12.23 在平行四边形模式上的多重加工 307



CNC 8070

(SOFT V03.0x)

12.24 在网格模式上的多重加工 308
12.25 任意多重加工 309

第 13 章 坐标变换

13.1 在斜面上的运动 313
13.2 运动学选择 (#KIN ID) 315
13.3 坐标系统 (#CS) (#ACS) 316
13.3.1 坐标系统定义模式 MODE 1 319
13.3.2 坐标系统定义模式 MODE 2 321
13.3.3 坐标系统定义模式 MODE 3 323
13.3.4 坐标系统定义模式 MODE 4 324
13.3.5 坐标系统定义模式 MODE5 325
13.3.6 坐标系统定义模式 MODE6 326
13.3.7 主轴的操作 (Huron 型) 329
13.4 如何组合多个坐标系 330
13.5 对平面的刀具补偿 (#TOOL ORI) 332
13.6 使用 RTCP (旋转式刀具中心点) 334
13.6.1 RTCP 功能的注意事项 338
13.7 刀具长度补偿 (#TLC) 339
13.8 运动学相关参数 340
13.9 丢失平面如何退回刀具 341

第 14 章 倾斜轴的角度变换

14.1 旋转角度变换激活和取消 345
14.2 冷冻角度变换 346
14.3 角度轴结构校验 347

第 15 章 CNC 变量

15.1 变量说明概述 349
15.1.1 从 PLC 访问数字值 351
15.1.2 在单通道系统访问变量 352
15.1.3 单通道系统的变量访问 354
15.2 相关普通机床参数 357
15.2.1 相关通道 359
15.3 轴相关机床参数 361
15.3.1 换档相关参数 364
15.4 jog 模式相关参数 367
15.5 "M" 功能相关参数 368
15.6 运动学相关参数 369
15.7 刀库相关参数 370
15.8 OEM 相关 参数 371
15.9 相关用户表格 372
15.10 刀具相关变量 374
15.10.1 仅用在程序段准备期间的变量 377
15.11 PLC 相关变量 378
15.12 Jog 模式相关变量 379
15.13 坐标相关变量 381
15.14 进给率相关变量 383
15.15 主轴速度相关变量 384
15.16 编程功能相关变量 385
15.17 独立轴相关变量 390
15.18 机床设置相关变量 391
15.19 其它变量 394
15.20 变量列表 （按首字母顺序） 398



第 16 章 编程语句与指令

16.1 编程语句 408
16.1.1 显示指令 . 在屏幕上显示错误 408
16.1.2 显示指令 . 在屏幕上显示警告 410
16.1.3 显示指令 . 在屏幕上显示信息 411
16.1.4 显示指令 . 定义图形区尺寸 412
16.1.5 使能和取消使能指令 413
16.1.6 相对机床参考零点 （原点）编程 414
16.1.7 定义和执行子程序 415
16.1.8 执行程序段和程序的指令 419
16.1.9 从动电子轴 421
16.1.10 轴屏蔽 422
16.1.11 修改通道轴的配置 424
16.1.12 修改通道主轴的配置 429
16.1.13 选择通道的主控主轴 432
16.1.14 主轴同步 433
16.1.15 为轴或主轴选择开环或闭环 437
16.1.16 垂直刀具轴选择 439
16.1.17 "C" 轴 : 激活主轴作 "C" 轴 440
16.1.18 "C" 轴 : 零件表面加工 441
16.1.19 "C" 轴 : 零件回转面加工 443
16.1.20 碰撞检测 445
16.1.21 关于手动干涉 447
16.1.22 样条 (Akima) 450
16.1.23 多项式插补 453
16.1.24 高速加工 454
16.1.25 加速度控制 456
16.1.26 坐标变换 458
16.1.27 宏定义 460
16.1.28 程序段重复 462
16.1.29 通道间的通信和同步 464
16.1.30 独立轴的运动 468
16.1.31 附加编程指令 472
16.2 流程控制指令 475
16.2.1 程序段跳转 ($GOTO) 475
16.2.2 条件执行 ($IF) 476
16.2.3 条件执行 ($SWITCH) 478
16.2.4 程序段重复 ($FOR) 479
16.2.5 条件程序段重复 ($WHILE) 480
16.2.6 条件程序段重复 ($DO) 481

第 17 章 探测固定循环

17.1 #PROBE 1. 刀具标定 484
17.1.1 测量或标定刀具长度 485
17.1.2 测量或标定刀具半径 488
17.1.3 测量或标定刀具半径和长度 490
17.2 #PROBE 2. 探针标定 493
17.3 #PROBE 3. 表面测量固定循环 496
17.4 #PROBE 4. 外拐角测量固定循环 500
17.5 #PROBE 5. 内拐角测量固定循环 503
17.6 #PROBE 6. 角度测量固定循环 506
17.7 #PROBE 8. 外拐角和角度测量固定循环 508
17.8 #PROBE 8. 孔测量固定循环 511
17.9 #PROBE 9. 凸台测量固定循环 514

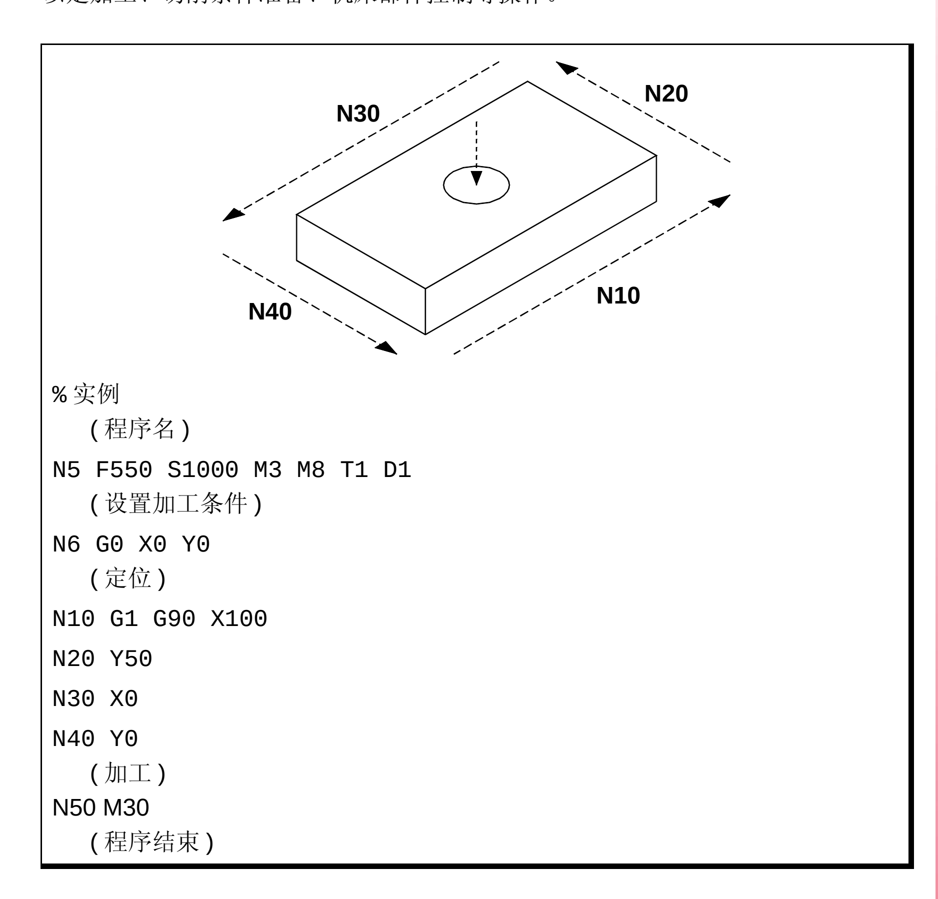


程序生成

1

1.1 程序结构

数控程序由一系列程序段或有组织的指令组成。子程序或程序体提供给数控装置加工工件所需的所有数据。每个程序段包含执行一个操作所需的所有功能或指令，可以是加工、切削条件准备、机床部件控制等操作。




```
% 实例
    ( 程序名 )
N5 F550 S1000 M3 M8 T1 D1
    ( 设置加工条件 )
N6 G0 X0 Y0
    ( 定位 )
N10 G1 G90 X100
N20 Y50
N30 X0
N40 Y0
    ( 加工 )
N50 M30
    ( 程序结束 )
```

数控程序可以由多个子程序和程序主体组成。

局部子程序

子程序是一系列程序段，一经确立后可以被其他子程序或程序体多次调用。编写子程序是随意的，而且必须在程序体之前定义。子程序通常用于定义在程序里重复出现多次的一组操作或运动。

子程序的开始以 "%L< 名称 >" 定义， < 名称 > 部分最长可以由 14 个字符的大写字母、小写字母或者数字组成 （不含空格）。调用子程序时区分大小写，名称必须按定义的格式正确书写 。子程序结束用 M17， M29 或者 #RET 定义。

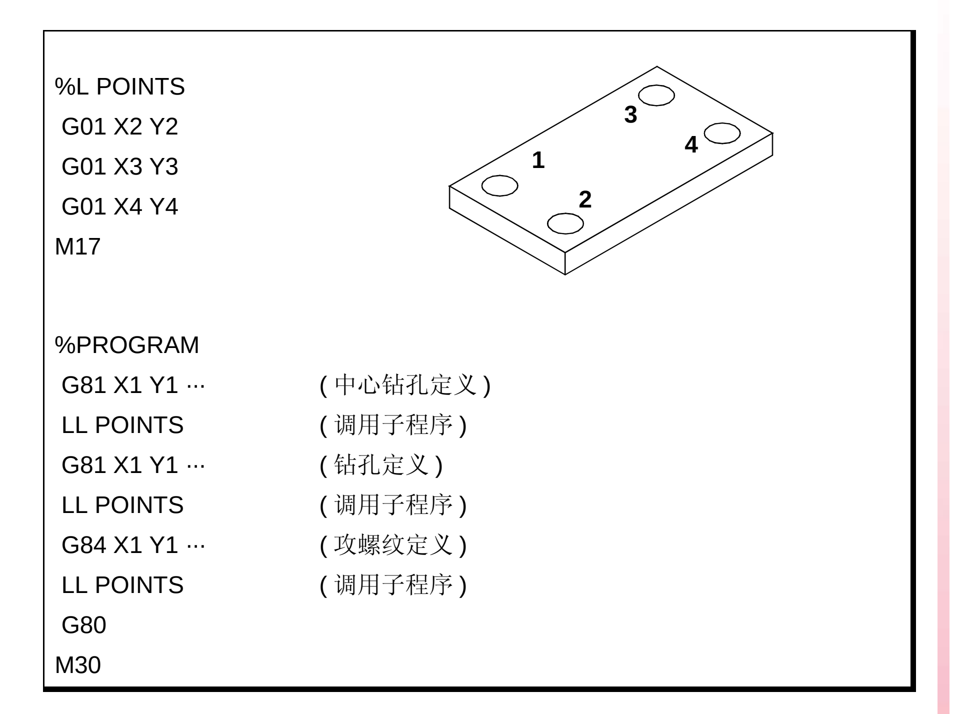
```
%L sub_name1          ( 子程序定义 )
 N10...
 N20...
 N30...
M17                   ( 子程序结束 )

%L sub_name2          ( 子程序定义 )
 N10...
 N20...
 N30...
M17                   ( 子程序结束 )
```

程序体

程序体的开始以 "%L< 名称 >" 定义， < 名称 > 部分 最长可以由 14 个字符的大写字母、小写字母或者数字组成 （不含空格）。不定义子程序时不需编程。

程序体结束由 "M02" 或 "M30" 定义，并且必须编写它们。




```
%L POINTS
 G01 X2 Y2
 G01 X3 Y3
 G01 X4 Y4
M17

%PROGRAM
 G81 X1 Y1 ···          ( 中心钻孔定义 )
 LL POINTS              ( 调用子程序 )
 G81 X1 Y1 ···          ( 钻孔定义 )
 LL POINTS              ( 调用子程序 )
 G84 X1 Y1 ···          ( 攻螺纹定义 )
 LL POINTS              ( 调用子程序 )
 G80
M30
```

1.2 程序段结构

组成子程序或者程序体的程序段可以用 ISO 代码或高级语言指令编写。每个程序段可以用其中一种语言编写，但是同一程序段中不能混用两种语言；一个程序可以由两种语言编写的程序段组成。允许出现空段 （空行）。

在任何一种语言中，也可能使用任何一种算法，关系式或逻辑式。

ISO 语言

该语言专门用于控制轴的运动，它给出运动数据与条件以及进给倍率和转速。

它包括：

- 用于控制运动的准备功能，决定几何形状和工作条件，例如直线和圆弧插补，螺纹加工，固定循环等。
- 控制切削条件的功能，例如控制进给率，主轴转速和其加速度。
- 刀具控制功能。
- 辅助功能，包括技术指令。
- 位置定义。

高级程序语言

高级程序语言为用户提供一系列控制语句，与其他语言的术语一样，例如：$IF，$GOTO， #MSG， # HSC 等。

这种类型的语言是：

- 编程指令。
- 在程序里也有构成循环和跳转的控制指令。
- 定义和调用子程序可以使用局部参数，局部变量即只在它所定义的子程序内有效的变量。

它也允许使用各种表达式 （算数、关系或逻辑）。

常数、参数、变量和算数表达式可以使用来自 ISO 程序段的，也可以是来自带 $ and # 的特殊指令的。

ISO 代码编程

ISO 代码功能指令由字母和数字组成。

- 字母有 "N"，"G"， "F"，"S"，"T"，"D"，"M"，"H"，"NR" 以及代表轴的字母。
- 数字包括阿拉伯数字从 "0" 到 "9"，符号 "+" 和 "-" 以及小数点 "."。另一方面，数字可以用一个参数、变量或算数表达式替换，在这一章稍后的部分 “第 1.4 节参数、常数和表达式” 中会说明。

程序段结构

程序段可以有以下的数据，但不是必须全部都包含。

/	N—	G— G—	X··· C—	F— S—	T—	M— H—	NR—	(—)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

这些数据没有预定的顺序，除了跳过程序段，它的段标志必须总是在程序段的开始编制。

1. 有条件的跳过程序段 "/"

   如果跳过程序段的标志有效，数控系统会跳过这些程序段接着执行随后的程序段，会保留这些指令但不执行。

为了预先计算运动路径，数控系统在执行一个程序段前读入多个程序段。跳过程序段的条件在程序段读入时检查。

2. 程序段标号 "N"

它被用作程序段标志，且在程序段调用或跳转时必须使用。在这种情况下，建议该程序段单独使用程序段号。它可用两种方式表示 :

• 字母 "N" 后接程序段号 (0-4294967295) 和符号 ":" (仅当该标志用作一个跳转程序段目标位置); 不必按特定或连续顺序。

  如果该标志不是一个跳转目标且没有编入符号 ":"，可以放到程序段的任何位置，不必在开头。

• "[<名称>]" 型标志，<名称> 部分最长可以14个字符的大小写字母以及数字组成 (不允许空格).

两种类型的数据可以在同一程序段编程。

3. 准备 G 功能

它决定几何形状和工作条件，例如直线和圆弧插补，倒角，固定循环等。

本章的 “1.2.1 准备功能 G 代码表”将介绍可用的 G 功能。

4. 点坐标 "X, Y, Z..."

它们决定轴的运动。

轴的名称用 1 或 2 个字符定义。第一个字母必须是字母 X - Y - Z - U - V - W - A - B - C 中的一个。第二个字符可选，且后面需要使用一个 1 到 9 之间后缀。照此，轴的名称可以在 "X, X1...X9,...C, C1...C9" 范围内。

运动用代表轴的字母跟该轴目标位置的坐标表示。

X100 Y34.54 X2 = 123,4 A5=78.532

依据单位不同 , 编程格式会是 :

• 公制， 格式 ±5.4 (5 位整数， 4 位小数)。

• 英制， 格式 ±4.5 (4 位整数， 5 位小数)。

5. "F" 和 "S" 功能

它们控制轴的进给率和主轴转速。

进给率用字母 "F" 加预定进给率数值表示。

6. 主轴转速用字母 "S" 加预定主轴转速值表示。

7. 刀具号 "T" 和刀具偏置 "D"

它选择刀具和刀具偏置用来执行程序编制的加工操作内容。刀具用字母 "T" 加刀具编号 (0-4294967295) 表示。刀具偏置用字母 "D" 加刀具偏置号表示 。

8. 辅助功能 "M" 代码和 "H" 代码

辅助功能可以控制机床元件的操作 , 例如主轴正反转，冷却液的开关等。

它用字母 "M" 或 "H" 加功能号码 (0-65535)。

9. 程序段重复次数 "NR"

它表明程序段执行的次数。

它只能在包含运动的程序段进行编辑。

如果程序段在模态固定循环影响之下，后面会重复和编程定义的次数一样多次。当程序编写 NR0，运动会执行，但是在每一次结束时模态固定循环不执行。

10. 程序段注释 "(...)"

给程序段加一个注释。当执行程序时，数控系统忽略此信息。

作为注释的信息必须写在圆括号 "(" 和 ")" 之间。不必在程序段尾；它可以在程序段中并且在同一程序段中可以有多个注释。

1.2.1 准备功能 G 代码表

下表给出数控系统中可用准备功能的清单。表中字母 "M", "D" 和 "V" 的含义如下：

M 表示模态指令；即一经编写，该 G 功能一直有效，直到再次编写了不兼容的 G 代码、M02、M30 或者急停或复位操作被执行，又或者数控系统掉电后重新上电。
符号 "!" 表示功能一直有效，即使在 M02， M30 或者复位操作和数控系统掉电又重新上电之后。

D 表示缺省值；即，在CNC启动、执行M02、M30、急停或复位操作后采用该指令。
符号 "?" 表示这些 G 功能缺省值取决于通用数控机床参数的设置。

V 表示此 G 功能在自动或手动模式下紧接着当前加工条件显示。

功能	M	D	V	含义	章节
G00	*	?	*	快速定位	6.1
G01	*	?	*	直线插补	6.2
G02	*		*	顺时针圆弧 （螺旋线）插补	6.3 / 6.6
G03	*		*	逆时针圆弧 （螺旋线）插补	6.3 / 6.6
G04			*	暂停	8.1
G05	*	?	*	圆角 (模态)	7.3
G06			*	圆心绝对坐标圆弧 (非模态)	6.3.5
G07	*	?	*	倒角 (模态)	7.1
G08			*	切于前一路径圆弧	6.4
G09			*	三点定义圆弧	6.5
G10	*	*		镜像取消	7.8
G11	*		*	X 轴镜像	7.8
G12	*		*	Y 轴镜像	7.8
G13	*		*	Z 轴镜像	7.8
G14	*		*	图像相对于编程方向镜像	7.8
G17	*	?	*	主平面 X-Y，纵轴为 Z 轴	3.1
G18	*	?	*	主平面 Z-X，纵轴为 Y 轴	3.1
G19	*		*	主平面 Y-Z，纵轴为 X 轴	3.1
G20	*		*	用两个方向定义主平面，纵轴	3.1.1
G30				极坐标原点预设	4.6
G31			*	临时极坐标原点移位至圆弧中心	6.3.4
G33	*		*	恒螺距螺纹切削	6.7
G36			*	自动半径过渡	7.4
G37			*	切入	7.6
G38			*	切出	7.7
G39			*	自动倒角过渡	7.5
G40	*	*		取消刀具半径补偿	9.1
G41	*		*	刀具半径左补偿	9.1
G42	*		*	刀具半径右补偿	9.1
G50	*	?		半圆角	7.2
G53	*			零点偏置取消	4.5
G54	!		*	绝对零点偏置 1	4.4
G55	!		*	绝对零点偏置 2	4.4
G56	!		*	绝对零点偏置 3	4.4
G57	!		*	绝对零点偏置 4	4.4
G58	!		*	绝对零点偏置 5	4.4
G59	!		*	绝对零点偏置 6	4.4
G60			*	方角 (非模态)	7.1
G61			*	受控圆角 (非模态)	7.3
G63	*		*	刚性攻螺纹	6.8
G70	*	?	*	英制编程	3.2
G71	*	?		公制编程	3.2
G72			*	比例缩放	7.10
G73	*		*	坐标旋转 (图形旋转)	7.9
G74			*	回参考点	2.4.2

功能	M	D	V	含义	章节
G80	*	*		取消固定循环	10.1.3
G81	*		*	钻削循环	10.2
G82	*		*	变步长往复式钻削循环	10.3
G83	*		*	恒步长往复式深孔钻削	10.4
G84	*		*	攻丝循环	10.5
G85	*		*	铰孔循环	10.6
G86	*		*	钻孔循环	10.7
G87	*		*	矩形型腔铣削循环	10.8
G88	*		*	圆形型腔铣削循环	10.9
G90	*	?		绝对坐标编程	3.3
G91	*	?	*	增量坐标编程	3.3
G92	!		*	坐标系建立	4.3
G93	*		*	加工时间 （单位：秒）	5.2.1
G94	*	?		进给率为每分钟进给	5.2.1
G95	*	?	*	进给率为每转进给	5.2.1
G96	*		*	恒线速度切削	5.3.1
G97	*	*		横转速度切削	5.3.1
G98	*	*		退回到起始平面	10.1.4
G99	*		*	退回到参考平面	10.1.4
G100			*	探测	8.6
G101	*			包含探针偏置	8.6.1
G102	*			不含探针偏置	8.6.1
G108	*	*		在程序段开始进给率混合	5.2.2
G109			*	在程序段末尾进给率混合	5.2.2
G112	*			变换轴的参数范围	8.5
G130	*		*	每轴加速度百分比	5.2.5
G131	*		*	所有轴加速度百分比	5.2.5
G132	*		*	每轴加加速度百分比	5.2.6
G133	*		*	所有轴加加速度百分比	5.2.6
G134	*		*	前馈百分比	5.2.7
G135	*		*	AC- 前馈百分比	5.2.8
G136	*		*	段间圆弧过渡	9.1.1
G137	*	*		段间直线过渡	9.1.1
G138	*		*	刀具补偿直接激活 / 取消	9.1.1
G139	*	*		刀具补偿间接激活 / 取消	9.1.1
G151	*	*	*	直径方式编程	3.4
G152	*			半径方式编程	3.4
G157	*		*	零点偏置排除	4.4.2
G158	*		*	增量零点偏置	4.4.1
G159	!		*	附加绝对零点偏置	4.4
G160			*	在直线模式上的多重加工	11.1
G161			*	在平行四边形模式上的多重加工	11.2
G162			*	在网格模式上的多重加工	11.3
G163			*	在整圆模式上的多重加工	11.4
G164			*	在圆弧模式上的多重加工	11.5
G165			*	在圆弧 - 弦模式上的多重加工	11.6
G170	*			Hirth 轴关	8.3
G171	*	*		Hirth 轴开	8.3
G180			*	原始设备制造商子程序	8.4
G181			*	原始设备制造商子程序	8.4
G182			*	原始设备制造商子程序	8.4
G183			*	原始设备制造商子程序	8.4
G184			*	原始设备制造商子程序	8.4
G185			*	原始设备制造商子程序	8.4
G186			*	原始设备制造商子程序	8.4
G187			*	原始设备制造商子程序	8.4
G188			*	原始设备制造商子程序	8.4
G189			*	原始设备制造商子程序	8.4
G192	*		*	转速限制	5.2.2
G193			*	插入进给率	5.2.2

功能	M	D	V	含义	章节
G196	*		*	切削点进给率恒量	5.2.3
G197	*	*		刀具中心进给率恒量	5.2.3
G198	*			设置下软行程极限	8.2
G199	*			设置上软行程极限	8.2
G200				专用手动干涉	6.9.2
G201	*			附加手动干涉激活	6.9.1
G202	*	*		附加手动干涉撤销	6.9.1
G261	*		*	圆心用绝对坐标 (模态)	6.3.5
G262	*	*		圆心相对于起点	6.3.5
G263	*		*	圆弧半径编程	6.3.2
G264	*		*	圆弧中心修正取消	6.3.6
G265	*	*		圆弧中心修正激活	6.3.6
G266			*	进给率修调 100%	5.2.4
G281			*	交互式中心钻循环	12.2
G282			*	交互式钻削循环 1	12.3
G283			*	交互式钻削循环 2	12.4
G284			*	交互式攻螺纹循环	12.5
G285			*	交互式铰削循环	12.6
G286			*	交互式镗削循环 1	12.7
G287			*	交互式矩形型腔循环	12.10
G288			*	交互式圆形型腔循环	12.11
G289			*	交互式简单型腔循环	12.9
G290			*	交互式平面铣削循环	12.17
G291			*	交互式矩形凸台循环	12.15
G292			*	交互式圆形凸台循环	12.16
G293			*	交互式点到点轮廓循环	12.18
G294			*	交互式轮廓循环	12.19
G295			*	交互式狭槽铣削循环	12.20
G296			*	交互式预制空腔型腔循环	12.12
G297			*	交互式镗削循环 2	12.8

1.3 高级语言编程

高级语言指令由控制指令 "#" 和流程控制指令 "$" 组成。

程序段结构

用高级语言编写的程序段可以由以下数据组成，但不必全部都有。

/	N—	高级语言指令	(—)
(1)	(2)	(3)	(4)

跳过程序段条件和程序段标识必须编写在程序段开始。

1. 有条件的跳过程序段 "/"

如果程序段跳过标记有效，数控系统将跳过有标志的程序段（不执行它们），并继续执行下一个程序段。

数控系统在执行程序前，提前读入多段程序，为了预先计算要行进的路径。在程序段读入时，检查程序段跳过条件。

2. 程序段号 "N"

当程序段用作参考的目标或跳过的程序段时必须编写程序段号。既然这样，推荐单段编写它。可以用于两种方法表示：

• 字母 "N" 后跟数字 (0-4294967295) 和符号 ":"（仅当标志用作程序段跳过的目标）；它不需要跟特殊指令或者是连续的。

  如果标志不是跳过目标并且没有编写 ":"，可以放在程序段的任何位置，不需要在程序段的开始。

• "[<名称>]"类型的标志，其中<名称>最长可达 14 个字符，字符可以是大写字母和小写字母也可以是数字（不允许是空白）。

两种类型的数据可以编写在同一个程序段。

3. 高级命令 "#—" 和 "$—"

高级命令由指令和流程控制指令组成。

• 指令前编写符号 "#"，并且每个程序段仅能编写一条指令。可用于执行各种各样的功能。

• 流程控制指令前编写符号 "$"，并且每个程序段仅能编写一个指令。用于构成循环和程序跳过。

参数和变量赋值也可以当作高级命令。

本手册 ***"16 Statements and instructions"*** 中介绍了所有可用的指令和指令。

4. 程序段注释 "(...)"

给程序段加入注释。当执行程序时，数控系统忽略此信息。

作为注释的信息必须放在圆括号 "(" 和 ")" 之间。不需要放在程序段末；可以放在程序段中间，也可以在同一程序段中有多个注释。

当用高级语言编程时，注释也可以用指令 "#COMMENT BEGIN" 和 "#COMMENT END" 定义。

1.4 参数，常数和表达式

常数、参数、变量和算数表达式可以使用来自 ISO 程序段的，也可以使用来自带 $ and # 的特殊指令的。

常数

它是固定值，不能通过编程修改；常数是十进制的数，只读类型，在程序里不能改变其值。

变量

CNC 有许多中间变量，它可以从用户程序、PLC 或界面访问。详见“第 15 章 CNC 变量”。

用户可以创建自己的变量，如下。

V.P.name - 程序局部用户变量。

V.P.name - 程序全局用户变量。

算数参数

参数的作用一般是被用户用作编写自己的程序。 CNC 有全局变量 （程序和子程序可用），局部变量 （仅在它被编写的程序和子程序可用）和通用参数。

这章的 ***"1.4.1 算术参数"*** 讲述如何使用参数。

算子

算子是一种说明要执行的数学或逻辑运算的符号。

手册中 ***"1.4.2 Operators and functions"*** 中介绍可用的多类算子和函数。

表达式

一个表达式是常数，参数，变量和算子的正确组合。

本章 ***"1.4.3 Expressions"*** 中介绍如何使用表达式。

1.4.1 算术参数

CNC 有三种类型的算数参数。每一种可用参数的范围在机床参数中被定义。

- 局部参数只能从它们被编写的程序或子程序访问。每个通道有 7 组局部参数。

  局部参数最大范围是从 P0 到 P99，典型使用范围是从 P0 到 P25。

  当参数在程序段中调用子程序时被使用，用到字母 A-Z （除了 Ñ）， "A" 同 P0 "Z" 同 P25。

- 全局变量可以在任何程序和该程序调用的子程序被访问。每个通道有一组全局变量。

  全局变量的最大范围是从 P100 到 P9999, 典型使用范围是 P100 到 P299。

- 通用参数可以从任何通道被访问。这些参数值可被所有通道共享。

  通用参数的最大范围是P10000 到P19999，典型使用范围是P10000 到P10999。

用户可以在编写自己的程序时使用这些参数。在执行过程中，CNC 会用赋给它们的值替换这些参数。

```
P0=0 P1=1 P2=20 P3=50 P4=3
P10=1500 P100=800 P101=30
...
GP0 XP0 YP0 SP10 MP4        ==>    G0 X0 Y0 S1500 M3
GP1 XP2 YP3 FP100           ==>    G1 X20 Y50 F800
MP101                       ==>    M30
```

编程

在用 ISO 代码编写的程序段中，可以用参数定义所有这些地址的值，它们是 "N", "G", "F", "S", "T", "D", "M", "H", "NR" 和轴的坐标。使用间接寻址 ，也可以用另外一个参数定义某个参数的标记，例如 "P[P1]", "P[P2+3]"。

在有 "#" 指令的程序段中，任何一个表达式的值可以用参数定义。

子程序里的参数

已定义子程序可以被主程序或另一个子程序调用；它们可以依次调用二级子程序，二级子程序又可以调用三级子程序，依此类推。该数控系统嵌套层的最大极限是 20 次。

局部参数

CNC 有全局参数 (从程序或任意子程序访问) 和局部参数 (仅从被编写的程序或子程序访问)。

局部参数可以被赋值给多个子程序 （参数嵌套层可达 7 层），子程序嵌套层数在 20 以内。不是所有子程序调用类型会改变嵌套层；仅仅 #PCALL， #CALL ，调用指令和功能指令 G180 到 G189。

全局参数

全局参数可以被程序和其通道的子程序共享。它们可以在程序和子程序的任何程序段被使用，不管它们所在的嵌套层。

通用参数

通用参数可以被程序和任何通道的子程序共享。它们可以在程序和子程序的任何程序段被使用，而不管它们所在的嵌套层。

1.4.2 算子和函数

算子是一种符号，用来说明要执行的算数或逻辑操作 。CNC 提供以下几种类型的算子。

算法

执行算数操作。

+	加法	P1 = 3+4	P1=7
-	减法 改变符号	P2 = 5-2 P2 = -[3+4]	P2=3 P2 = -7
*	乘法	P3 = 2*3	P3=6
/	除法	P4 = 9/2	P4=4.5
MOD	模数或余数	P5 = 5 MOD 2	P5=1
**	幂	P6 = 2**3	P6=8

在运算中，当使用参数或变量存储结果，加法、减法、乘法和除法运算时可以按下述方式使用。

+=	复合加法	P1 += 3	P1=P1+3
-=	复合减法	P2 -= 5	P2=P2-5
*=	复合乘法	P3 *= 2	P3=P3*2
/=	复合除法	P4 /= 9	P4=P4/9

关系式

用于做比较。

==	等于	P1 == 4
!=	异于，不等于	P2 != 5
>=	大于等于	P3 >= 10
<=	小于等于	P4 <= 7
>	大于	P5 > 5
<	小于	P6 < 5

进制比较

用于在常数之间和 / 或算术表达式之间的二进制比较 .

<table>
<tr><td>&</td><td>二元与</td><td>P1 = P11 & P12</td></tr>
<tr><td>|</td><td>二元或</td><td>P2 = P21 | P22</td></tr>
<tr><td>^</td><td>异或</td><td>P3 = P31 ^ P32</td></tr>
<tr><td>INV[...]</td><td>反</td><td>P4 = INV[P41]</td></tr>
</table>

如果常数或算术表达式的结果是十进制数，小数点部分被忽略。

逻辑

用于在两个条件间做逻辑比较。

*	逻辑与	$IF [P11 == 1] * [P12 >=5]
+	逻辑或	$IF [P21 != 0] + [P22 == 8]

每个条件应该用括号括起来，否则，由于在两个算子中的优先权可能会致使不希望的比较。

布尔代数

TRUE	真	$IF V.S.VAR == TRUE
FALSE	假	$IF V.S.VAR == FALSE

三角函数

SIN[...]	正弦	P1 = SIN[30]	P1 = 0.5
COS[...]	余弦	P2 = COS[30]	P2 = 0.866
TAN[...]	正切	P3 = TAN[30]	P3 = 0.5773
ASIN[...]	反正弦	P4 = ASIN[1]	P4 = 90
ACOS[...]	反余弦	P5 = ACOS[1]	P5 = 0
ATAN[...]	反正切	P6 = ATAN[1]	P6 = 45
ARG[...]	反正切 y/x	P7=ARG[-1,1]	P7=225

必须记住以下类型的函数：

- 在 "TAN" 函数中，自变量不能取这些值 ...-90º, 90º, 270º...
- 在 "ASIN" 和 "ACOS" 中，自变量必须在 ±1 以内。
- 有两个函数可用于计算反正切值 :

"ATAN" 返回的结果在 ±90º 之间。

"ARG" 返回的结果在 0º 和 360º 之间。

算术运算

ABS[...]	绝对值	P1 = ABS[-10]	P1 = 10
SQR[...]	平方	P2 = SQR[4]	P2 = 16
SQRT[...]	开方	P3 = SQRT[16]	P3 = 4
LOG[...]	以 10 为底的对数	P4 = LOG[100]	P4 = 2
LN[...]	以 e 为底的对数	P5 = LN[100]	P5 = 4.6051
EXP[...]	"e" 指数	P6 = EXP[1]	P6 = 2.7182
DEXP[...]	以 10 为底的指数	P6 = DEXP[2]	P7 = 100

使用该类函数，请注意如下几条 :

- 在 "LN" 和 "LOG" 函数中，自变量必须大于 1。
- 在 "SQRT" 函数中，自变量必须为整数。

其它函数

INT[...]	返回整数	P1 = INT[4.92]	P1 = 4
FRACT[...]	返回小数	P2 = FRACT[1.56]	P2 = 0.56
ROUND[...]	四舍五入圆整	P3 = ROUND[3.12] P4 = ROUND[4.89]	P3 = 3 P4 = 5
FUP[...]	返回整数 +1(如果自变量为整数，则返回其本身)。	P5 = FUP[3.12] P6 = FUP[9]	P5 = 4 P6 = 9
EXIST[...]	检查输入的变量是否存在。	$IF EXIST[P1] $IF EXIST[P3] == FALSE	

在 "EXIST" 函数中，程序 "$IF EXIST[P1] == TRUE" 与 "$IF EXIST[P1]" 等效。

1.4.3 表达式

表达式是运算符号、常数、参数和变量的正确组合。

运算符号的优先级和组合方式决定了表达式的运算方式：

优先级由高到低	组合方式
函数 , - (负号)	从右到左
** (指数), MOD (余数)	从左到右
* (乘法 , 逻辑与), / (除法)	从左到右
+ (加法 , 逻辑或), - (减法)	从左到右
关系运算符	从左到右
& (AND),^ (XOR)	从左到右
| (OR)	从左到右

可使用括号明确表达式的求解关系。使用多余的括号既不会产生错误，也不会影响系统执行速度。

P3 = P4/P5 - P6 * P7 - P8/P9

P3 = [P4/P5] - [P6 * P7] - [P8/P9]

算术表达式

其结果是数字值。它由算术计算、二进制运算符以及常数、参数、变量组合而成。该类表达式也可用于向参数和变量赋值 :

P100 = P9   P101 = P[P7]   P102 = P[P8 + SIN[P8*20]]

P103 = V.G.TOOL

V.G.FIXT[1].X=20   V.G.FIXT[1].Y=40   V.G.FIXT[1].Z=35

关系表达式

其结果是 TRUE 或 FALSE。它是关系和逻辑运算符、算术表达式、常数、参数和变量的组合。

... [P8==12.6] ...

该式判断 P8 的值是否等于 12.6。

... ABS[SIN[P4]] > 0.8 ...

该式比较 P4 的正弦值的绝对值是否大于 0.8。

... [[P8<=12] + [ABS[SIN[P4]] >=0.8] * [V.G.TOOL==1]] ...

机床概述

2

2.1 轴的命名

数控系统允许厂商可以选择多达 28 根轴（必须通过设置机床参数，将其定义为直线、旋转等），并且不限制它们的编程方式，同时它们都可以内插值替换。

DIN 66217 标准轴的命名：

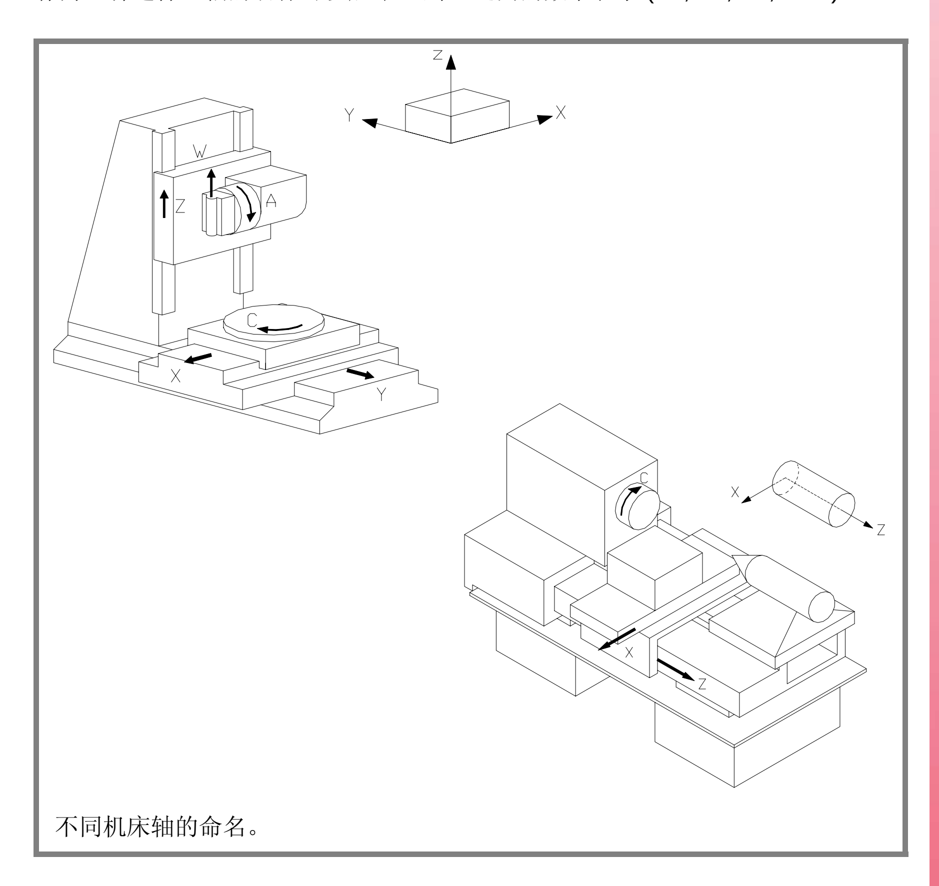
X-Y-Z 机床主轴。 X-Y 轴构成主工作平面，Z 轴平行于主轴且垂直于 XY 平面。

U-V-W 附加轴，分别平行于 X-Y-Z 。

A-B-C 旋转轴，旋转中心分别平行于 X-Y-Z

然而，机床制造商也可能以不同的名字命名轴。

作为一种选择，轴的名称可以后跟 1 到 9 之间的数字表示 (X1, X3, Y5, A8...).



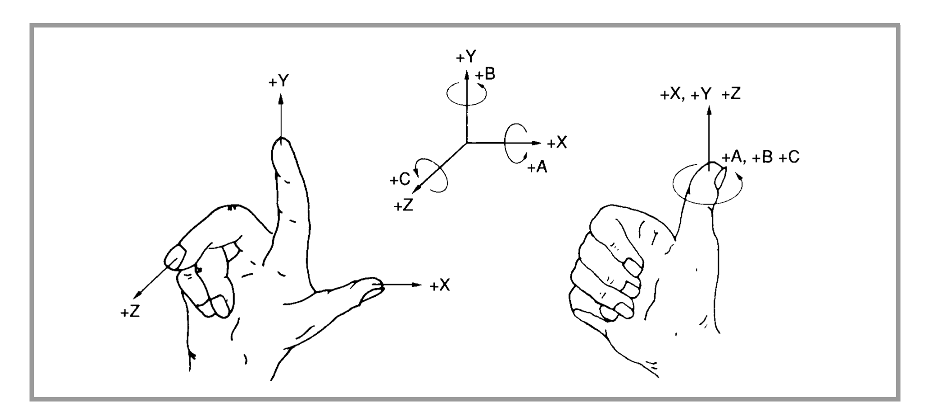
不同机床轴的命名。



右手法则

X-Y-Z 轴的方向可以用右手法则容易地记住。（如下图所示）。

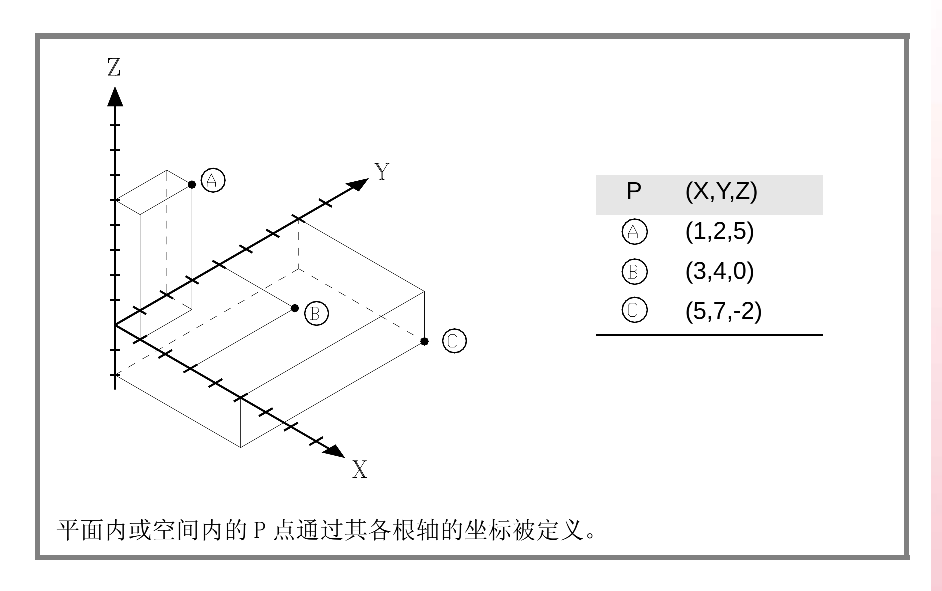
当拇指指向直线轴的正方时，手握住旋转轴方向。旋转轴正方向由手指所指的方向决定。



2.2 坐标系统

CNC 系统的目的之一就是控制轴的运动和位置，所以坐标系统必须可以在二维平面或三位空间里确定不同目标终点的坐标。

主坐标系由 X-Y-Z 轴组成。轴相互垂直交于原点，原点用作各种各样点的参考点。



平面内或空间内的 P 点通过其各根轴的坐标被定义。

其它类型的轴例如附加轴和旋转轴也可以是坐标系统的一部分。

2.3 参考系统

机床可以使用以下参考系。

这是机床的坐标系统，并由机床的制造厂商设置。

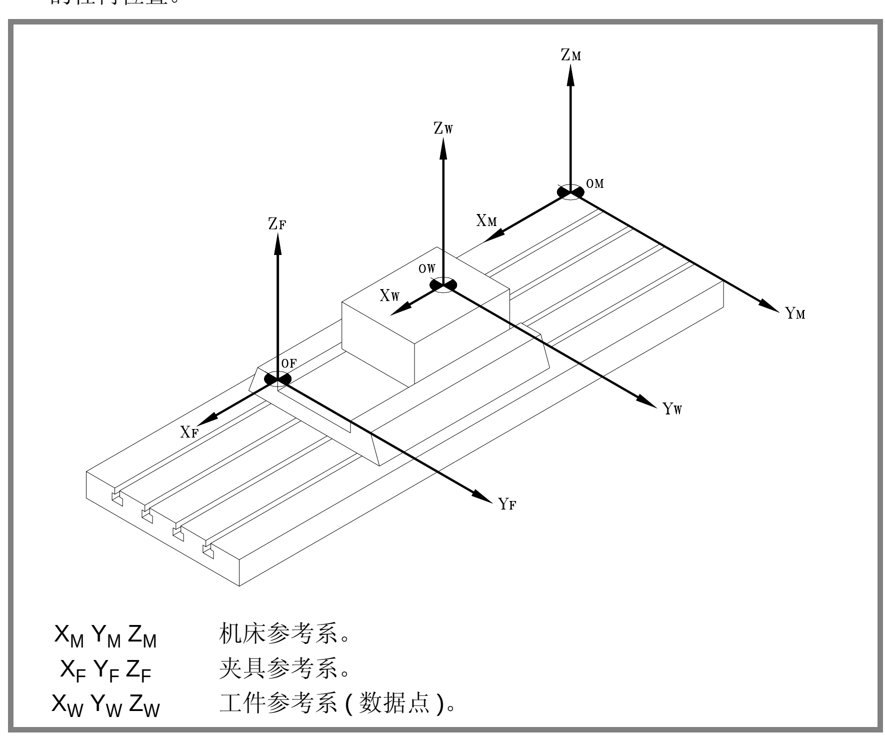
- 夹具参考系。

它建立与所用夹具相关的坐标系统。它通过程序激活，可由操作者设置在在机床的任何位置。

当机床使用多个夹具时，每一个都可以有与其关联的参考系统。

工件参考系 (数据点)。

它建立的坐标系统与被加工工件相关。它通过程序激活，可由操作者设置在零件的任何位置。



X_M Y_M Z_M	机床参考系。
X_F Y_F Z_F	夹具参考系。
X_W Y_W Z_W	工件参考系 (数据点)。

2.3.1 参考系原点

不同参考系的位置是由各自的原点位置决定的。

O_M **机床原点**

机床参考系的原点，由机床制造厂商设置。

O_F **夹具原点**

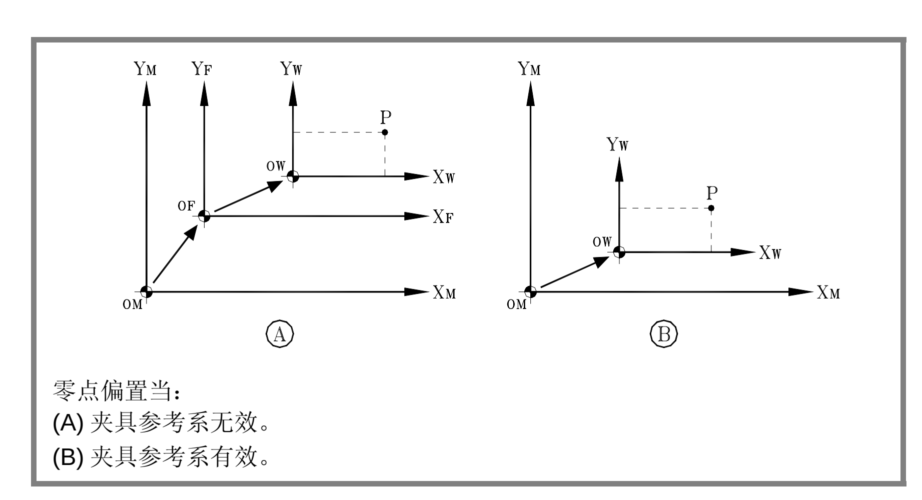
用在夹具参考系的原点。其位置由操作者使用 " 夹具偏置 " 定义，而且是相对于机床零点的位置。

"夹具偏置"可以通过编程或数控系统的控制面板设置，与操作说明书里介绍的一样。

O_W **工件原点**

它是工件参考系统的原点。它的位置由操作者使用用 " 零点偏置 " 设置，相对于：

- 夹具偏置，如果夹具参考系有效。当变换夹具参考系时，数控系统依据涉及的新的夹具零点更新工件原点位置。
- 机床原点，如果夹具参考系无效。激活夹具参考系时，数控系统通过使其相对于新的夹具零点来更新工件零点位置。
- 根据操作手册所介绍的那样， " 零点偏置 " 可以通过编程或从 CNC 面板设置。



零点偏置当:
(A) 夹具参考系无效。
(B) 夹具参考系有效。



2.4 原点搜索

2.4.1 定义 " 原点搜索 "

用于与系统同步的操作。当系统失去原点位置时必须执行该操作。(例如关掉机床)。

为了执行 “原点搜索”，机床制造商设置了特殊的点：机床零点和机床参考点。

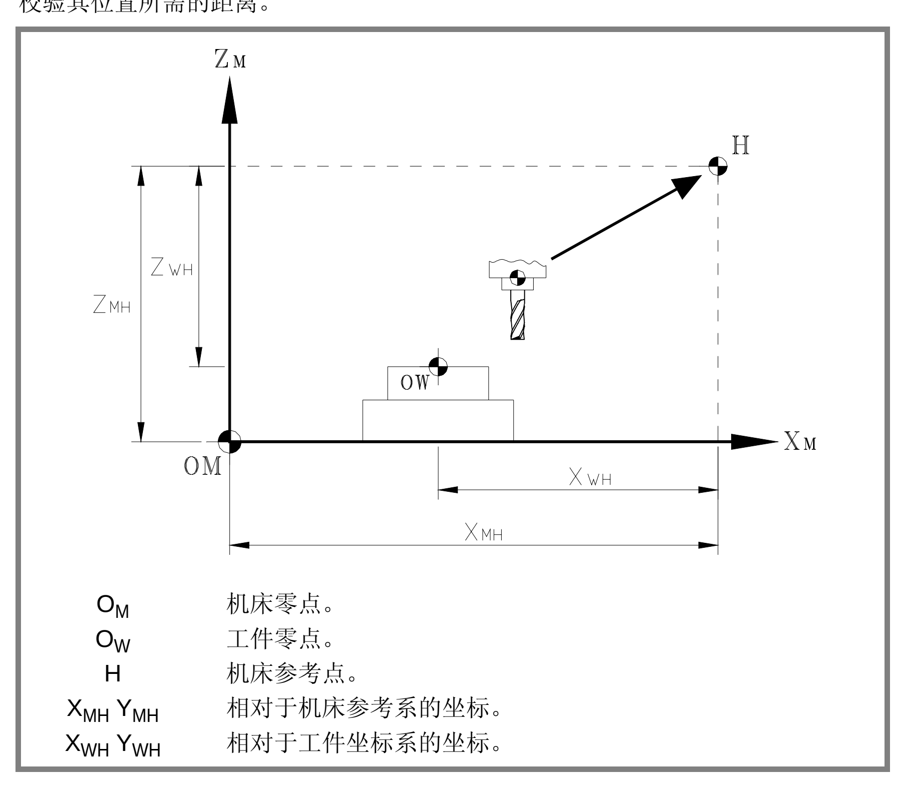
- 机床零点。

  机床参考系的原点。

- 机床参考点。

  系统同步的地方它是物理点（除了当机床使用 I_0 距离码参考标记或绝对反馈）。它可以设置在机床的任何位置。

执行 “原点搜索”时，轴运动到机床参考点，数控系统采用机床制造商赋予该点的坐标值 （相对于机床零点）。当使用 I_0 距离码参考标记或绝对反馈时，轴只会运动校验其位置所需的距离。



O_M	机床零点。
O_W	工件零点。
H	机床参考点。
X_{MH} Y_{MH}	相对于机床参考系的坐标。
X_{WH} Y_{WH}	相对于工件坐标系的坐标。



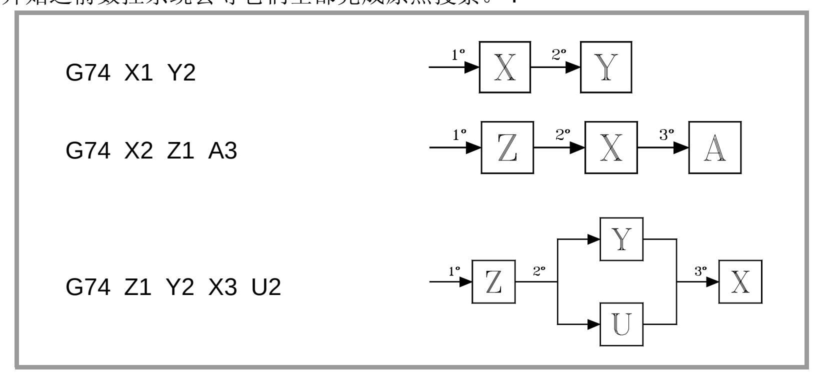
编写 " 原点搜索 " 时，不会取消夹具偏置和零点偏置；因此，坐标在有效参考系中显示。

另一方面，如果每次一根轴在手动方式 （不是 MDI 方式）下执行 “原点搜索”，有效偏置被取消，且坐标相对于机床零点显示。

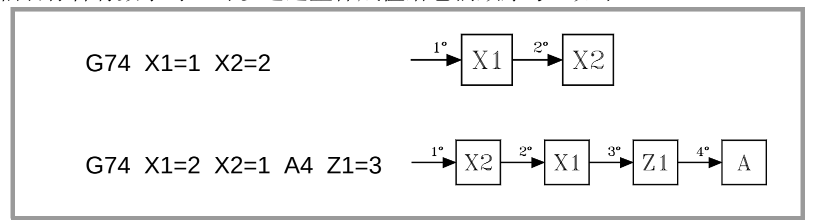
2.4.2 “原点搜索”编程

当编写 “原点搜索”时，各轴以操作者设定的顺序依次原点搜索。不是所有的轴必须包括在 “原点搜索”里，仅是那些需要搜索原点的。

原点搜索用 G74 编程，编程时后接要完成原点搜索的轴和指示原点搜索的顺序号。如果同样的顺序号被赋给几根轴，这些轴同时执行原点搜索，并且在下次的原点搜索开始之前数控系统会等它们全部完成原点搜索。 I



当轴名称含有数字时，可以通过整体赋值给它们顺序号，如下。



主轴原点搜索

使用位置控制主轴时，它可以像其他轴一样包含在 “原点搜索”中。这种情况下，主轴原点搜索总是和第一根轴一起执行，而不管其定义的顺序。

使用关联子程序

如果机床制造商关联原点搜索子程序到 G74，这个功能可以在程序段里单独编写，数控系统会自动执行相关的子程序 。 [G.M.P. "REFPSUB (G74)"]。

使用一个子程序时，“原点搜索”可以按照上述介绍的那样正确执行。







坐标系统

3

3.1 平面选择 (G17/G18/G19/G20)

平面选择决定那些轴构成工作平面 / 三面体和哪根轴与工件的纵向轴一致。执行如下这些操作时需要平面选择：

- 圆弧和螺旋插补。
- 倒角和圆角。
- 切出和切入。
- 加工固定循环。
- 刀具半径和长度补偿。

这些操作除了刀具长度补偿，仅可以在有效工作平面执行。另一方面，刀具长度补偿仅可以应用在纵向轴方向。

编程

使用这些 G 功能工作平面可以通过编程选择：

G17	X-Y 主平面，Z 轴纵向轴并且垂直于主平面。
G18	Z-X 主平面， Y 轴纵向轴并且垂直于主平面。
G19	Y-Z 主平面， X 轴纵向轴并且垂直于主平面。
G20	工作平面 / 三面体和纵向轴。

和使用指令：

#TOOL AX	纵向轴选择。

G17, G18 和 G19 功能注意事项：

在此 G 功能中提到 X，Y 和 Z 轴，并不代表这些轴必须使用这些名字；习惯使用第一坐标系的轴。

因此，当选择 G17， G18 或 G19 时，意义如下。

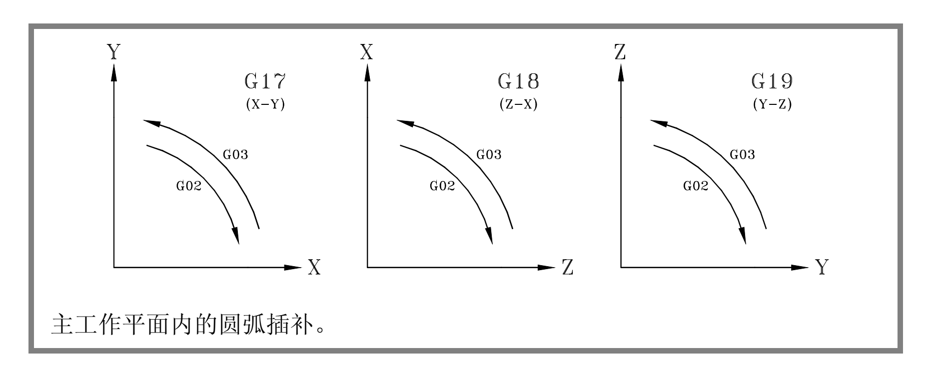
G17	选择该组坐标系第一和第二坐标构成的主平面。垂直轴 (螺旋的) 或纵向轴和坐标系第三轴一致。
G18	选择该组坐标系第三和第一坐标构成的主平面。垂直轴 (螺旋的) 或纵向轴和坐标系第二轴一致。
G19	选择该组坐标系第二和第三坐标构成的主平面。垂直轴 (螺旋的) 或纵向轴和坐标系第一轴一致。

垂直轴 （螺旋）是执行螺旋线插补的轴。纵向轴是应用刀具长度补偿的轴。当编写 G17， G18 和 G19 时垂直轴和纵向轴是相同的。

主平面和主轴

主平面可以通过 G 功能 G17，G18 和 G19 编程选择。主平面由坐标系的前两根轴定义。第三根轴与纵向轴方向一致，对于 G 功能 G17，G18 和 G19 纵向轴与平面的垂直轴方向一致。





主工作平面内的圆弧插补。

这些功能可以在程序的任何位置编写，不需要在程序段单独编写。

功能特性

G17，G18， G19 和 G20 功能是模态的，相互不兼容。

上电，执行 M02 或 M30 以及急停或复位后，数控系统采用机床厂商 [G.M.P. "IPLANE"] 设置的默认值 G17 或 G18。



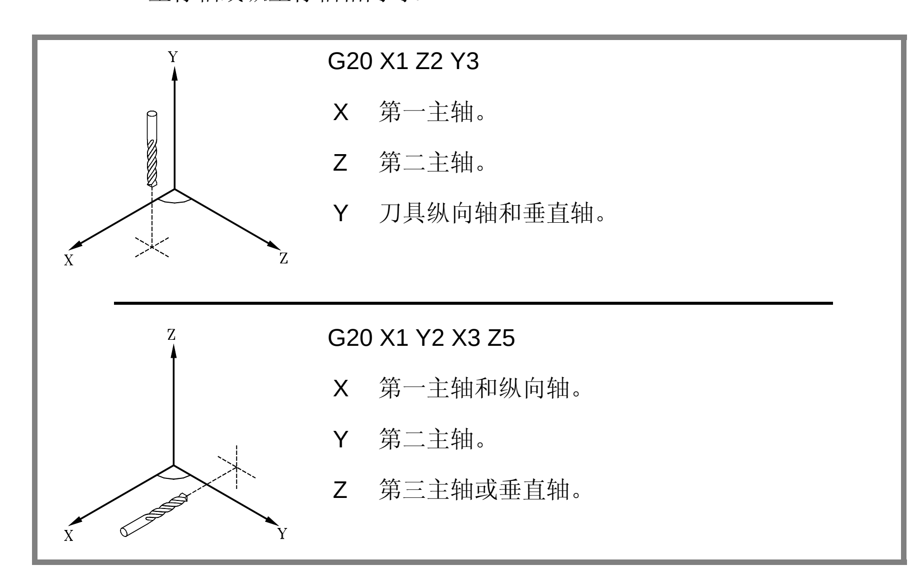
3.1.1 编写两方向定义工作平面 (G20)

除了主平面外，任何其它由通道中前三根轴形成的工作平面 / 三面体可以用 G20 进行定义。

编程

工作平面的定义是通过选择横轴，纵轴，垂直轴和刀具的垂直向轴实现。通过将下列参数赋予与 G20 一起编程的轴进行选择。。

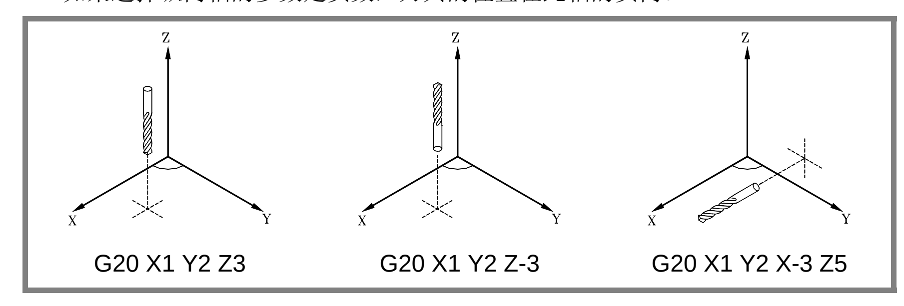
- "1" 给工作平面的第一轴 (横坐标轴)。
- "2" 给工作平面的第二轴 (纵坐标轴)。
- "3" 给刀具的垂直向轴，也是平面的垂直轴（螺旋）如果参数 5 没有被定义。
- "4" 保留。
- "5" 给工作平面的垂直轴；如果没有定义，同纵向轴。仅当纵向刀具轴与横坐标轴或纵坐标轴相同时。



注意事项

当用 G20 选择纵向轴时，刀具方向可以根据编程符号被确定。

- 如果选择纵向轴的参数是正数，刀具的位置在此轴的正向。
- 如果选择纵向轴的参数是负数，刀具的位置在此轴的负向。







3.1.2 垂直向刀具轴选择

刀具的垂直向轴可以用指令 "#TOOL AX" 选择，这条指令允许选择任何机床轴作为新的纵向轴。

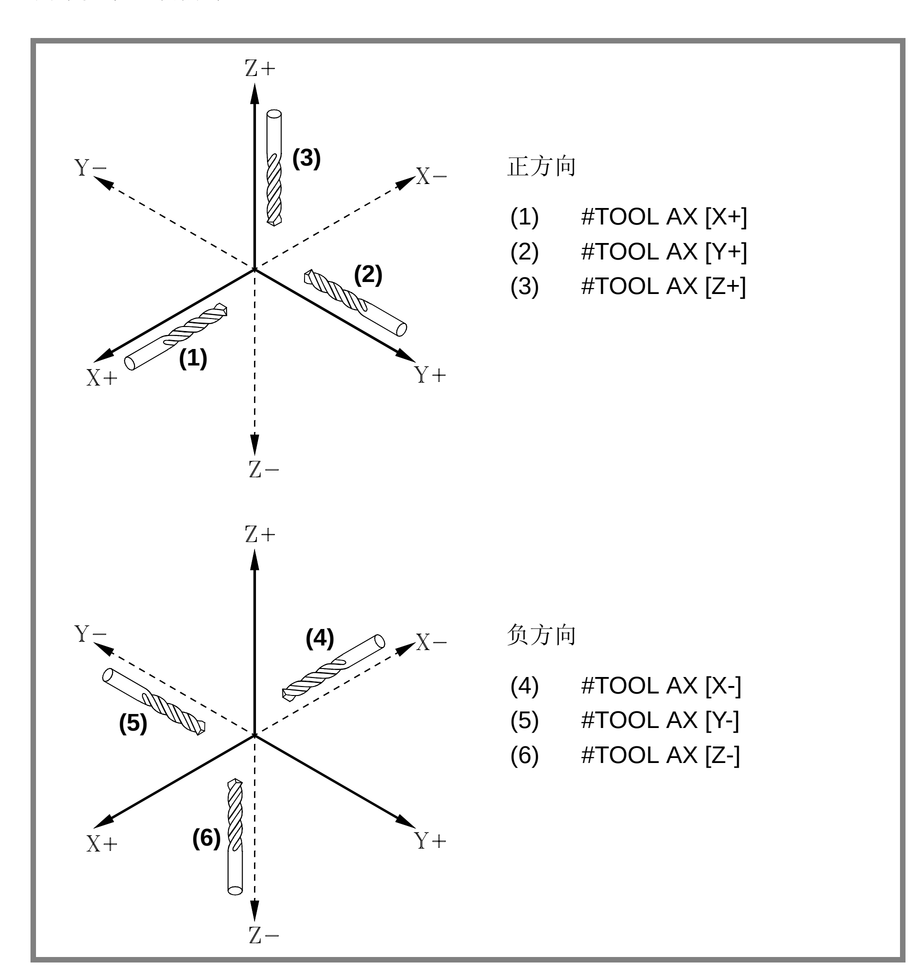
编程

刀具的垂直向轴用指令 "#TOOL AX [<axis><sign>]" 定义，在指令中：

- 参数 < 轴 > 规定新刀具的垂直向轴 。
- 参数 < 符号 > 表示刀具的方向。

+	正号表示刀具放置在轴的正方向。
-	负号表示刀具放置在轴的负方向。

两个参数必须编写。





3.2 公制 (G71) 和英制 (G70) 编程

轴的位移和进给率单位可以定义为毫米或英寸。度量单位系统可以用以下功能代码通过编程选择：

G70	以英寸为单位编程。
G71	以毫米为单位编程。

两种功能代码可以在程序的任何位置编写；不需要单独程序段编写。

操作

执行其中的一个G功能代码之后，数控系统在此后的所有程序段中采用相应的单位。如果没有编写其中任何一个 G 功能代码，数控系统用机床厂商设置的默认单位系统 [机床参数 "INCHES" 设定]。

当改变单位系统，数控系统转换当前运行的进给率执行新的单位制。

```
...
G01 G71 X100 Y100 F508  ( 公制单位编程 )
                        ( 进给率 : 508 mm/ 分钟 )
...
G70                     ( 改变单位制 )
                        ( 进给率 : 20 英寸 / 分钟 )
...
```

功能特性

G70 和 G71 功能是模态指令，互不兼容。

上电，执行 M02 或 M30 以及急停或复位后，数控系统采用机床厂商设置的默认单位系统 （G70 或 G71）。[由机床参数 "INCHES" 设置]

3.3 绝对编程 (G90) 或增量编程 (G91)

每个点的坐标可以用绝对坐标值（相对于坐标原点）或增量坐标值（相对于前一位置）。坐标的类型可以通过编写以下功能代码选择：

G90	绝对坐标值编程。
G91	增量坐标值编程。

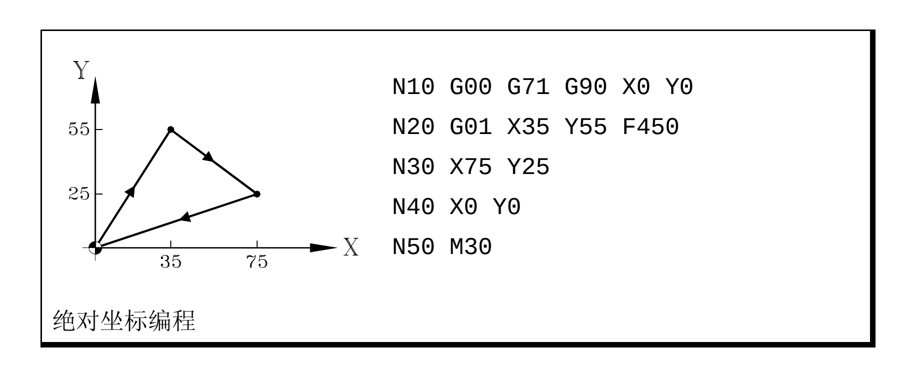
两种功能代码都可以在程序的任何部分编写；不需要单独程序段编写。

操作

执行其中的一个 G 功能代码之后，数控系统在此后的所有程序段中采用相应的坐标值模式。如果没有编写其中任何一个 G 功能代码，数控系统用机床厂商设置的默认的坐标模式。 [由机床参数 "ISYSTEM" 设置].

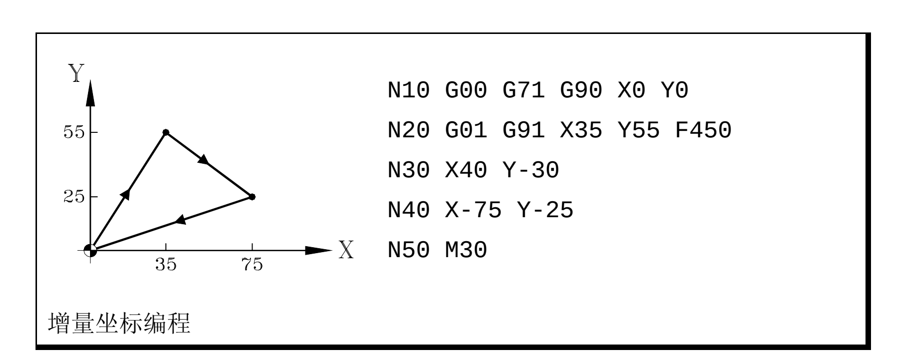
根据当前坐标模式 (G90/G91)，点的坐标按照如下方式定义 :

- 绝对坐标编程时 (G90)，点的坐标相对于坐标系原点，通常是工件零点。



绝对坐标编程

- 增量坐标编程时 (G91)，点的坐标相对于前一刀具位置。数值前的符号说明运动的方向。



增量坐标编程

功能特性

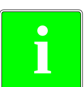
功能 G90 和 G91 是模态指令，互不兼容。

上电，执行 M02 或 M30 以及急停或复位后，数控系统采用机床厂商设置的默认坐标模式 （G90 或 G91）。[由机床参数 "ISYSTEM" 设置]





3.4 半径编程 (G152) 或直径编程 (G151)



以下功能针对车床。直径编程仅用于机床厂商允许的轴。(DIAMPROG=YES)。

半径编程还是直径编程可以通过编写以下功能代码选择：

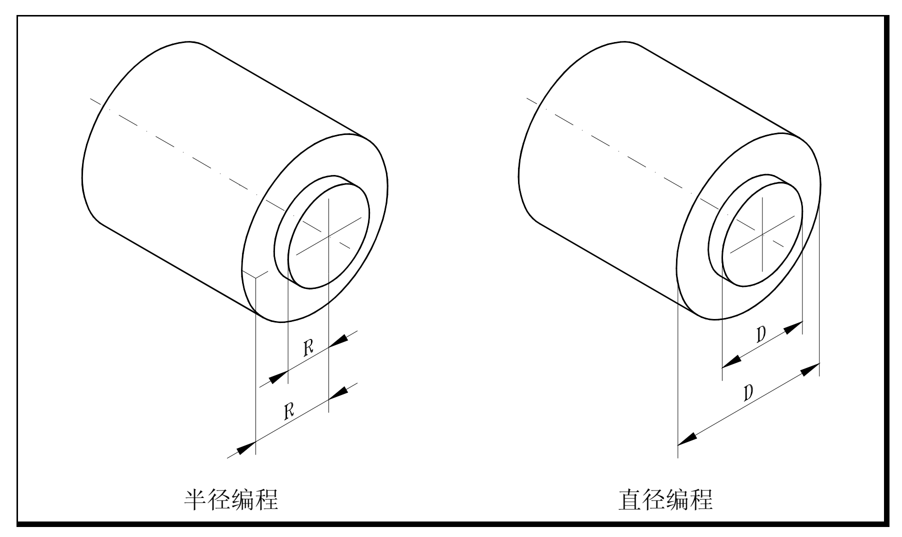
G151 直径编程。

G152 半径编程。

这些功能代码都可以在程序的任何部分编写；不需要单独程序段编写。

操作

执行这些指令中的一个后，数控系统确定以后程序段的编程模式。



当转换编程模式时，数控系统改变相应轴的显示方式。

功能特性

功能 G151 和 G152 是模态指令，互不兼容。

上电，执行 M02 或 M30 以及急停或复位后，如果坐标轴其中任何一根的机床参数 DIAMPROG 设置为 YES，数控系统采用功能 G151。

3.5 编程坐标

3.5.1 笛卡尔坐标

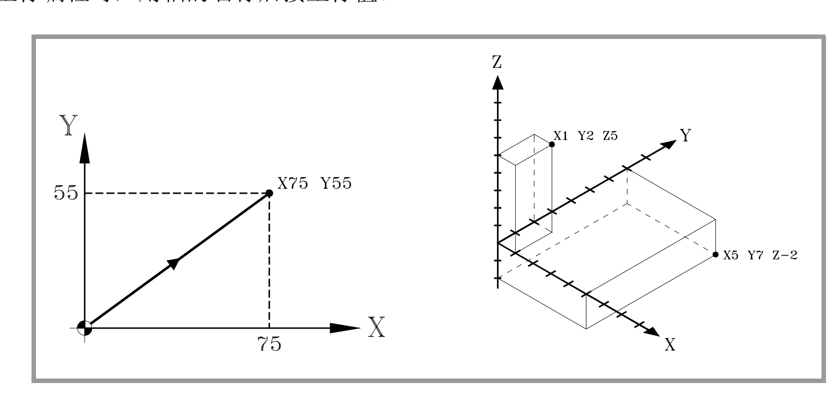
坐标根据笛卡尔坐标系编程。这种坐标系统由平面上的两根轴和空间的三根或更多的轴组成。

位置值定义

在笛卡尔坐标中的点的位置通过不同轴的坐标来确定。坐标采用绝对编程或者增量编程，公制单位或英制单位。

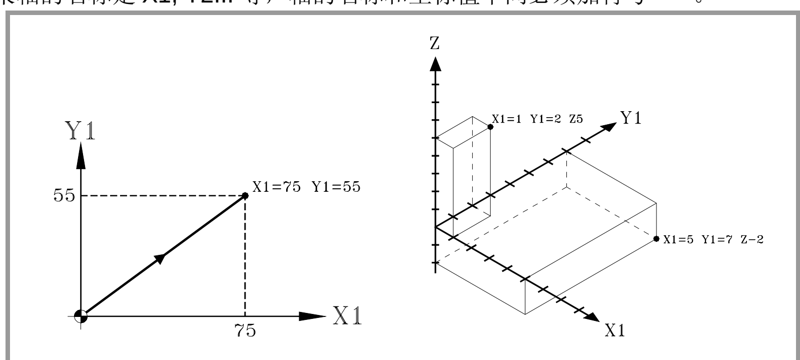
标准轴 (X...C)

坐标编程时，用轴的名称后接坐标值。



数字命名的轴 (X1...C9)

如果轴的名称是 X1, Y2... 等，轴的名称和坐标值中间必须加符号 "=" 。





3.5.2 极坐标

当出现圆弧或角度图形元素时，平面上不同点的坐标用极坐标表示更为方便。

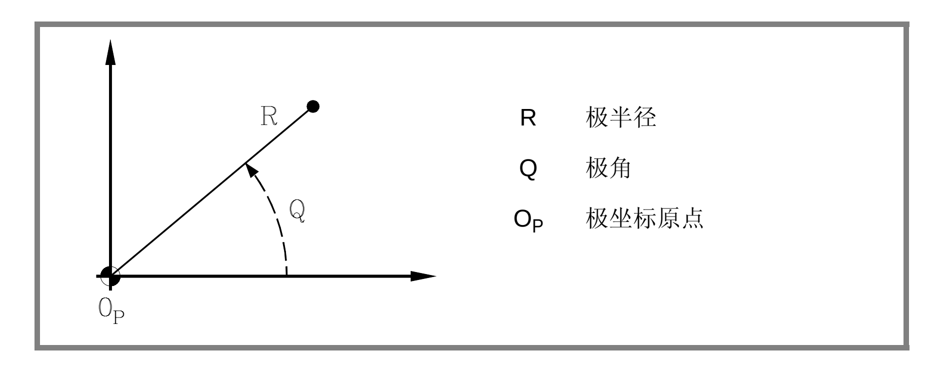
这种类型坐标需要参考点作为 “极坐标原点”，它是极坐标系的起始点。

位置值定义

各种点的位置通过定义极半径 “R” 和极角 “Q” 给出，如下:

半径　　极坐标原点和该点间的距离。

角度　　由横坐标轴和连接极坐标原点与该点的直线形成的角。



极半径单位可以是毫米或英寸，极角单位必须是度。

两个值都可以是绝对坐标 (G90)，也可以是增量坐标 (G91)。

- 在 G90 方式下，"R" 和 "Q" 值是绝对模式。赋给极半径的值必须始终是绝对值或者零。
- 在G91方式下，"R" 和 "Q" 值是增量模式。虽然 "R" 值可以编写负值，但是当在增量坐标编程时，极半径赋于的值必须始终是正值或零。

当编写 大于 360º 的 "Q" 值时，系统将用它除以 360 度。因此，Q420 与 Q60 相同，Q-420 与 Q-60 相同。

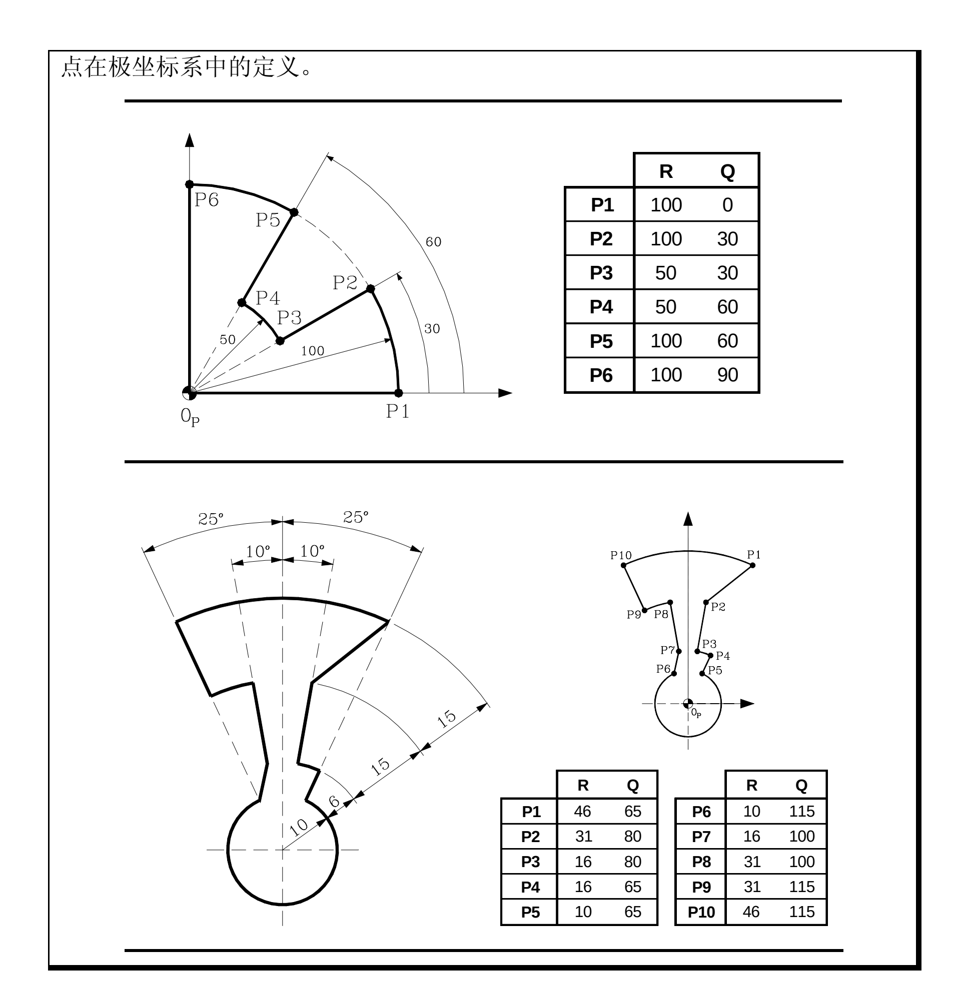
极坐标原点设定

极坐标原点可以通过编程时使用功能 G30 进行选择。如果没有选择，采用当前坐标系的坐标原点 （工件零点）作为极坐标原点。见 “4 原点选择”。

在下列情况时已选择的极坐标原点位置被修正:

- 当改变工作平面时，数控系统采用新的工件原点作为极坐标原点。
- 上电，执行M02或M30以及急停或复位后，数控系统采用工件原点作为新的极坐标原点。

编程实例



	R	Q
P1	100	0
P2	100	30
P3	50	30
P4	50	60
P5	100	60
P6	100	90

	R	Q
P1	46	65
P2	31	80
P3	16	80
P4	16	65
P5	10	65

	R	Q
P6	10	115
P7	16	100
P8	31	100
P9	31	115
P10	46	115



原点选择

4

对于数控系统，可以在机床参考系下或者应用偏置编写运动轨迹，这样便可使用相对于夹具或工件的参考系，从而不需要改变程序当中每个点的坐标。

有三种类型的偏置：

- 夹具偏置
- 零点偏置
- PLC 偏置

这几种偏置可以同时有效。所以，使用的坐标系统可以通过全部这些有效的偏置来定义。

夹具偏置

夹具偏置定义为机床参考零点和夹具零点间的距离。

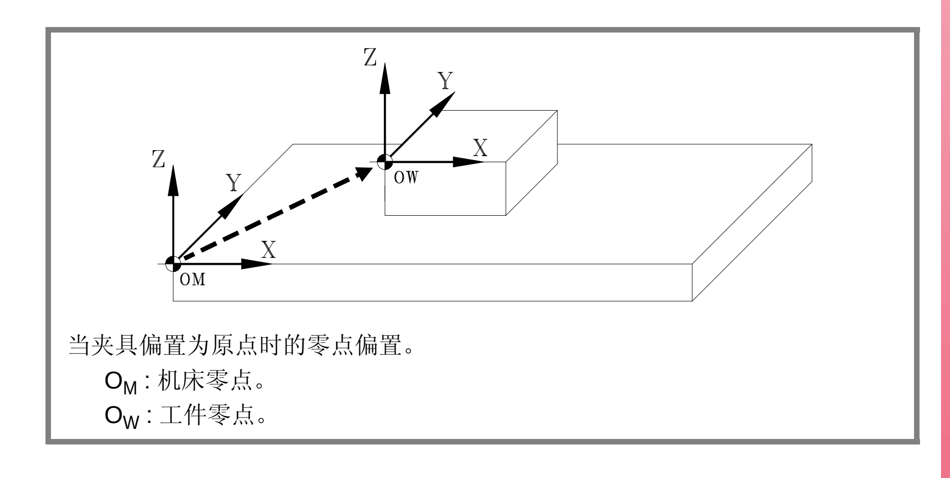
当机床使用多个夹具时，偏置允许选择使用特定的夹具。

零点偏置

零点偏置定义为夹具零点和工件零点间的距离。如果夹具零点无效（无夹具偏置），零点偏置从机床零点起测量。

零点偏置可以用两种方法实现：

- 通过坐标预置，数控系统采用编程坐标为当前轴的位置。
- 通过使用绝对或增量方式的零点偏置，数控系统采用通过选择偏置建立的新工件零点。



当夹具偏置为原点时的零点偏置。

O_M : 机床零点。

O_W : 工件零点。

PLC 偏置

由 PLC 控制的一种特殊偏置，用于矫正由于扩张等引起的偏差。

这种偏置始终由效，即使在相对于机床零点编程时。



4.1 相对于机床零点编程

机床零点是机床参考系的原点。相对于机床零点的运动用指令 #MCS 和 #MCS ON/OFF 编程。

零点偏置和坐标变换

当执行相对于零点的运动时，数控系统忽略有效的偏置 （除了 PLC 偏置），运动学和笛卡尔变换；所以，运动在机床参考系下执行。一旦运动结束，数控系统恢复曾经运行的偏置、运动学和笛卡尔变换。

编程的运动不允许是极坐标，或其它的变换，例如镜像，坐标 （图形）旋转或比例缩放。当功能#MCS起作用时，也不允许建立新原点功能例如G92, G54-G59, G158, G30 等。

刀具半径和长度补偿

在相对机床零点运动期间，刀具半径和长度补偿同样被取消。数控系统采用的编程坐标是相对于刀具基点，而不是刀尖。

单位：米制或英制

当相对机床参考点运动时，忽略由使用者选择的 G70 或 G71 功能确定的单位制（英制 / 公制）。采用数控系统中预先设定的单位 (参数 INCHES) ；数控系统上电时被采用。这些单位制在定义坐标，进给速率，转速时被采用。

#MCS 指令

这条指令可以加在任何包含运动的程序段中，它在机床参考系执行。

```
G00 X30 Y30
G92 X0 Y0           ( 坐标预置 )
G01 X20 Y20
#MCS X30 Y30        ( 相对于机床零点的运动。偏置取消 )
G01 X40 Y40         ( 偏置恢复 )
G01 X60 Y60
M30
```




#MCS ON 和 #MCS OFF 指令

#MCS ON 和 #MCS OFF 指令作用是激活和取消机床参考系；因此，编写在它们之间的运动在机床参考系执行。

```
G92 X0 Y0            (坐标预置)
G01 X50 Y50
#MCS ON              (相对机床零点编程开始，偏置取消)
 G01 ...
 G02 ...
 G00 ...
#MCS OFF             (相对于机床零点编程结束，偏置恢复)
```

两条指令都必须在单独段编写。

4.2 夹具偏置

用夹具偏置，可以选择要用到的夹具系统 （当有多个夹具）。当应用新的夹具偏置时，数控系统采用新选择的夹具上设置的点作为新的夹具零点。

定义

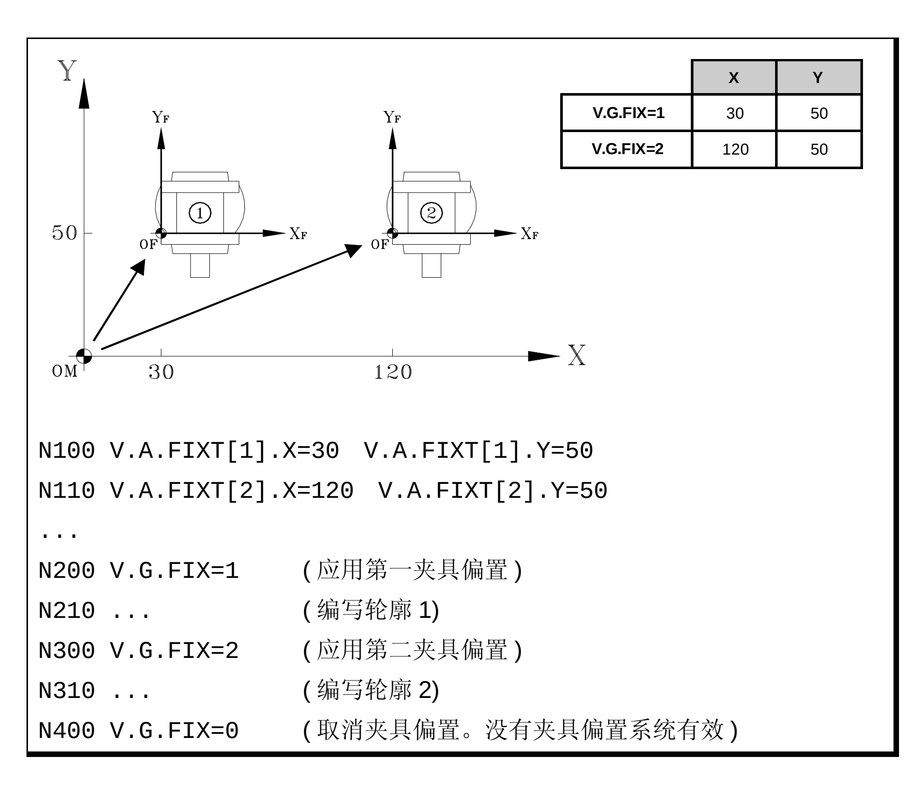
为了应用夹具偏置，必须预先设置。数控系统有一个表格，操作者可以定义多达十个不同的夹具偏置。表格里的数据可以通过以下方式定义：

- 手动从数控面板定义 (如操作手册所介绍的)。
- 通过编程赋予相应的值 ("n" 偏置和 "Xn" 轴的) 给变量 "V.A.FIXT[n].Xn"。

激活

一旦在表中定义夹具偏置，可以通过编程赋值给变量 "V.G.FIX" 和待应用的偏置号进行激活。

每次只能激活一个机床偏置；因此，当应用夹具偏置时，会取消前一个夹具偏置。赋值 "V.G.FIX=0" 将取消有效夹具偏置。



	X	Y
V.G.FIX=1	30	50
V.G.FIX=2	120	50

```
N100 V.A.FIXT[1].X=30  V.A.FIXT[1].Y=50
N110 V.A.FIXT[2].X=120  V.A.FIXT[2].Y=50
...
N200 V.G.FIX=1      ( 应用第一夹具偏置 )
N210 ...            ( 编写轮廓 1)
N300 V.G.FIX=2      ( 应用第二夹具偏置 )
N310 ...            ( 编写轮廓 2)
N400 V.G.FIX=0      ( 取消夹具偏置。没有夹具偏置系统有效 )
```

注意事项

夹具偏置本身不能引起任何轴的运动。



特性

在机床上电时，数控系统采用关机时的夹具偏置。另一方面，夹具偏置不受 M02 和 M30 功能以及数控系统复位影响。

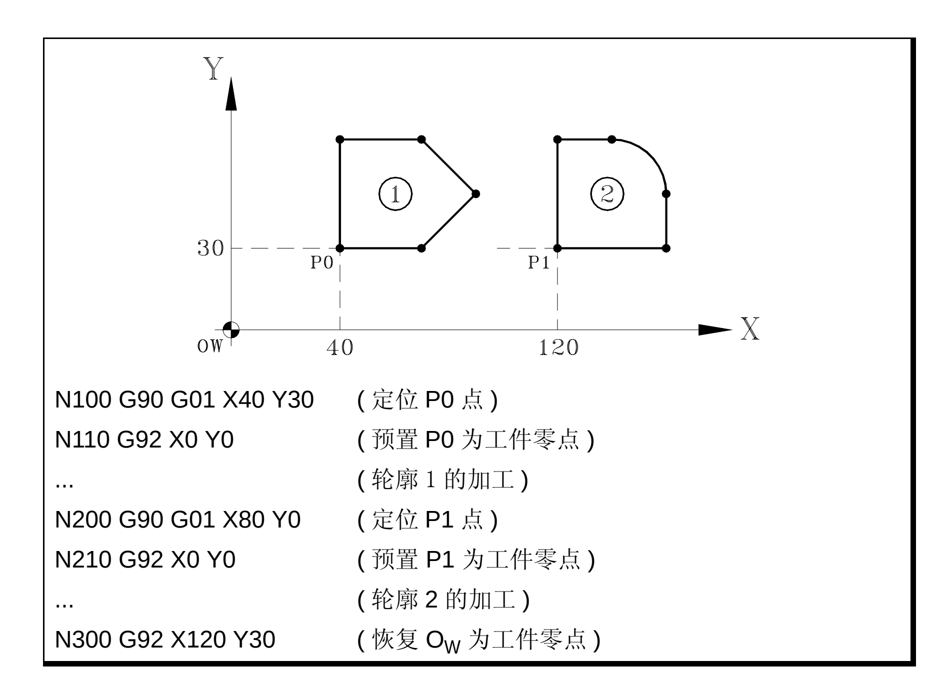




4.3 坐标预置 (G92)

用功能指令 G92 完成坐标预置，可以应用在机床的任何轴。

当预先调整一个坐标，数控系统认为该轴坐标从 G92 设置的轴的当前位置后开始编程。其余的没有被 G92 定义的轴不受该预置的影响。



N100 G90 G01 X40 Y30	(定位 P0 点)
N110 G92 X0 Y0	(预置 P0 为工件零点)
...	(轮廓 1 的加工)
N200 G90 G01 X80 Y0	(定位 P1 点)
N210 G92 X0 Y0	(预置 P1 为工件零点)
...	(轮廓 2 的加工)
N300 G92 X120 Y30	(恢复 O_W 为工件零点)

注意事项

坐标预置本身不能使任何轴运动。

当在手动方式下回零轴时，该轴的预置被取消。

功能特性

G92 是模态指令，预置值保持有效直到该预置被取消 (用另一个预置，零点偏置或 G53 取消)。

上电时，数控系统采用关机时的有效坐标预置。另一方面，坐标预置不受 M02 和 M30 功能以及数控系统复位影响。

4.4 零点偏置 (G54-G59/G159)

用零点偏置，可以在不同的位置设置工件零点。当应用零点偏置时，数控系统采用由可选零点偏置定义的点作为新的工件零点。

定义

为了应用零点偏置，必须预先定义它。数控系统具有零点偏置表，可以定义多达 20 个不同的零点偏置。表的数据可以通过以下方式定义：

- 手动从数控面板定义 (如操作手册所介绍的)。
- 编程赋相应值给变量 "V.A.ORGT[n].Xn" （其中的偏置 "n" 和轴 "Xn"）。

激活

一旦在表中定义零点偏置，就可以通过编程指令 G54 ～ G59 和 G159 激活。

G54 ˜G59 绝对零点偏置

应用零点偏置表中的前六个零点偏置。作用同编写 G159=1 到 G159=6。

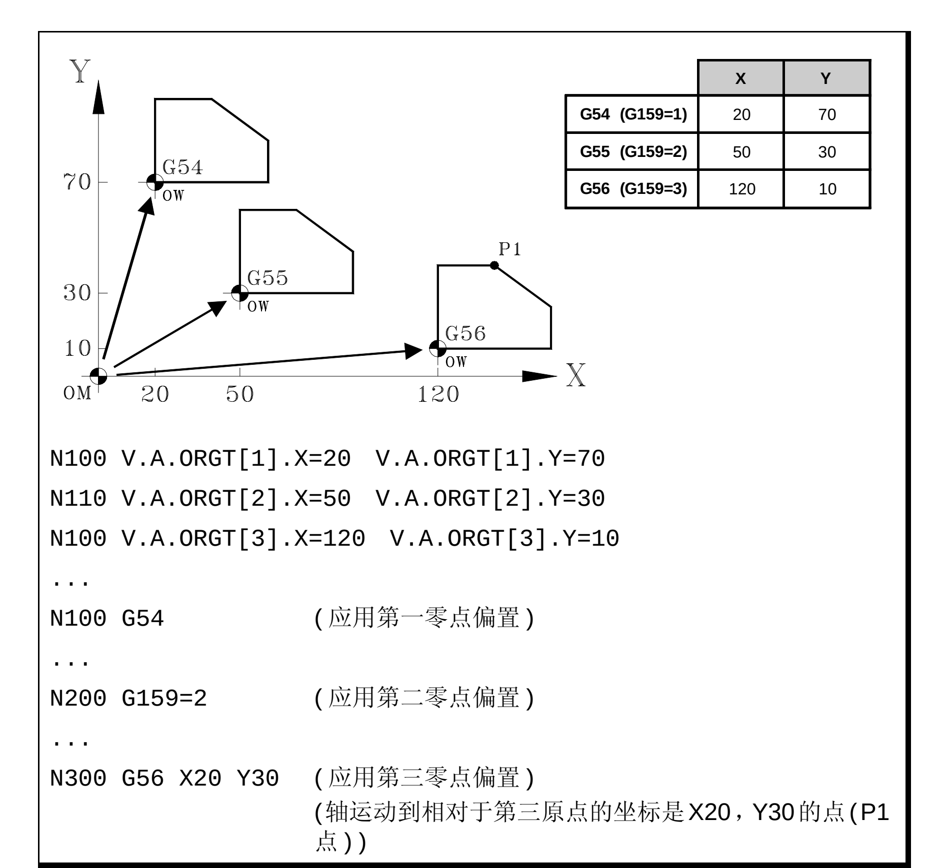
G54	应用第 1 零点偏置 (G159=1)。
G59	应用第 6 零点偏置 (G159=6)。

G159 附加绝对零点偏置

应用在表中定义的任意零点偏置。

前六个零点偏置同 G54 到 G59 编程。

G159=2	应用第 2 零点偏置。
G159=11	应用第 11 零点偏置。



	X	Y
G54 (G159=1)	20	70
G55 (G159=2)	50	30
G56 (G159=3)	120	10

```
N100 V.A.ORGT[1].X=20  V.A.ORGT[1].Y=70
N110 V.A.ORGT[2].X=50  V.A.ORGT[2].Y=30
N100 V.A.ORGT[3].X=120  V.A.ORGT[3].Y=10
...
N100 G54            (应用第一零点偏置)
...
N200 G159=2         (应用第二零点偏置)
...
N300 G56 X20 Y30    (应用第三零点偏置)
                    (轴运动到相对于第三原点的坐标是X20，Y30的点(P1点))
```

每次只能有一个零点偏置有效；因此，当应用零点偏置时，将取消前一个零点偏置。编写 G53 时，将取消当前有效的零点偏置。

可选零点偏置功能可以在程序的任意程序段编写。当给一个程序段添加路径信息时，零点偏置在执行编写的运动之前被应用。

注意事项

坐标预置本身不能使任何轴运动。

当在手动方式下回零轴时，该轴的绝对零点偏置被取消。

功能特性

功能 G54, G55, G56, G57, G58, G59 和 G159 是模态指令，彼此之间以及与 G53，G92 互不兼容。

上电时，数控系统采用关机时运行的零点偏置。零点偏置不受 M02 和 M30 功能以及数控系统复位影响。

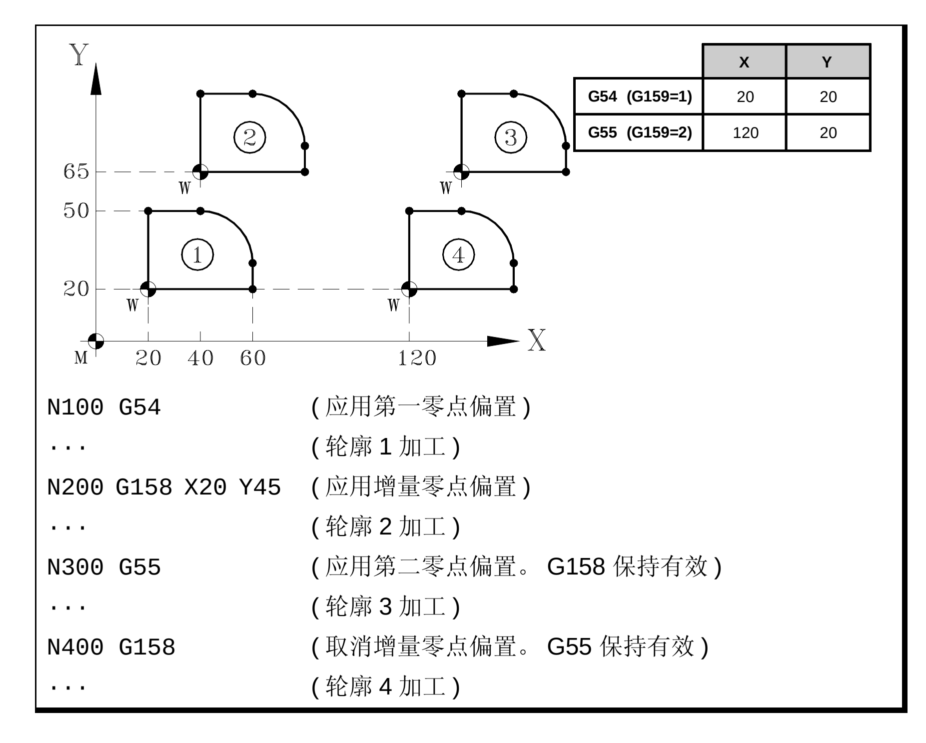
4.4.1 增量零点偏置 (G158)

当应用增量零点偏置时，每次数控系统加它到绝对零点偏置之上。

编程

增量零点偏置通过使用功能 G158 编程定义，G158 后跟应用于每根轴的零点偏置值。

取消增量零点偏置，在程序段中编写功能 G158，没有轴的零点偏置值。只取消特定轴的增量零点偏置，编写增量零点偏置 "0" 值给那些轴 。

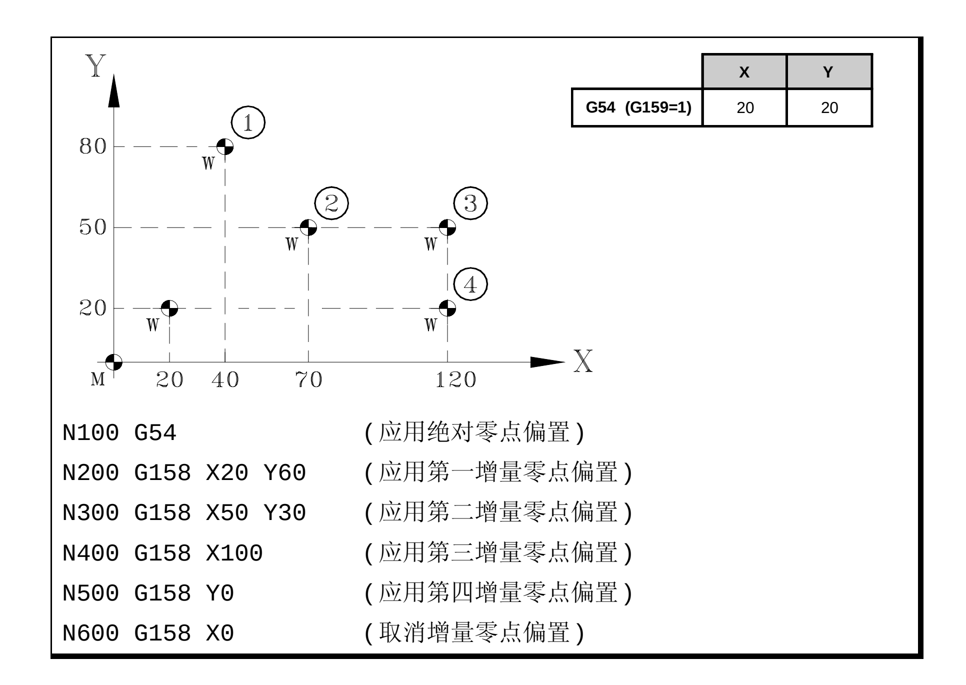


	X	Y
G54 (G159=1)	20	20
G55 (G159=2)	120	20

```
N100 G54              (应用第一零点偏置)
...                   (轮廓 1 加工)
N200 G158 X20 Y45     (应用增量零点偏置)
...                   (轮廓 2 加工)
N300 G55              (应用第二零点偏置。 G158 保持有效)
...                   (轮廓 3 加工)
N400 G158             (取消增量零点偏置。 G55 保持有效)
...                   (轮廓 4 加工)
```

每根轴每次仅有一个增量零点偏置；因此，在轴上应用增量零点偏置时将取消该轴原来的增量零点偏置。其它轴的偏置不受影响。





应用新的绝对零点偏置 (G54-G59 or G159) 后，增量零点偏置不会被取消。

注意事项

增量零点偏置本身不能使任何轴运动。

当在手动方式下回零轴轴，该轴的增量零点偏置被取消。

功能特性

功能 G158 是模态。

上电时，数控系统采用关机时运行的增量零点偏置。另一方面，增量零点偏置不受 M02 和 M30 功能以及数控系统复位影响。

4.4.2 零点偏置排除的轴 (G157)

选择排除的轴将在下一个绝对零点偏置中允许不应用偏置。在应用零点偏置之后，编写的轴的排除被取消，如果要再次应用它，必须再次编程。

激活

轴的排除必须用G157后加被排除的轴来编程，并且轴的值指明该轴被排斥(<轴>=1)还是不被排除 (< 轴 >=0)。

通过在功能 G157 之后单独编写受排除影响的轴，也可以激活排除。

排除和零点偏置可以在同一程序段编写。如果是那样的话，排斥将在应用零点偏置之前被激活。

```
G55
    ( 对所有的轴应用第二零点偏置 )
```

```
G157 X Z
    ( 激活对 X，Z 轴的零点偏置排除 )
G57
    ( 除了 X，Z 轴应用第四零点偏置，X，Z 轴保持以前的零点偏置 )
...
G159=8
    ( 对所有轴应用第八零点偏置 )
```

```
G59 G157 Y
    ( 除了 Y 轴应用第六零点偏置，Y 轴保持以前的零点偏置 )
...
G54
    ( 对所有的轴应用第一零点偏置 )
```

零点偏置排除的轴不影响当前的零点偏置。如果一个轴被排斥，当应用新零点偏置时，数控系统保持那根轴的本来状态。

注意事项

零点偏置排除的轴不影响坐标预置或是增量零点偏置，这些总是应用到所有的轴。同样的，夹具偏置和 PLC 偏置也不受影响。

功能特性

功能 G157 是模态指令，功能保持有效，直到应用绝对零点偏置。

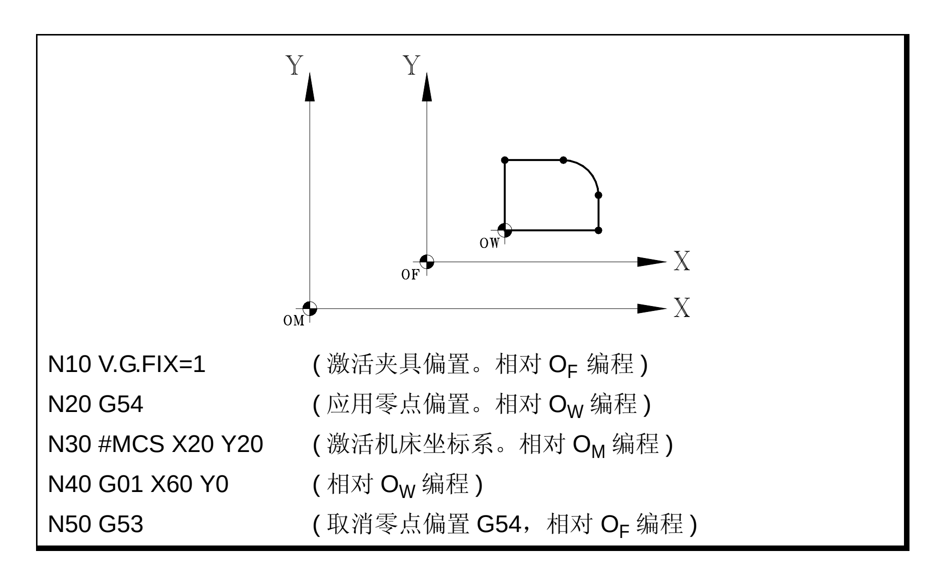
上电或急停之后，数控系统不采用任何轴的排除。

4.5 零点偏置取消 (G53)

执行功能 G53 取消当前零点偏置，其可以是由坐标预置 （G92）或是零点偏置产生的，零点偏置包括增量偏置和定义的轴的排除。也可由探针操作取消零点偏置。

夹具偏置和 PLC 偏置不受此功能影响。

与指令 #MCS 和 #MCS ON/OFF 相反，它们总是执行相对于机床零点的运动，功能 G53 允许执行相对于夹具零点 （它是有效的）的运动。



功能 G53 可以编写在程序的任何程序段。当给一个程序段附路径信息，在执行编写的运动之前，偏置或预制被取消。

注意事项

功能 G53 本身不使任何轴的运动。

功能特性

功能 G53 是模态指令，与功能 G92、零点偏置和探针相驳斥。



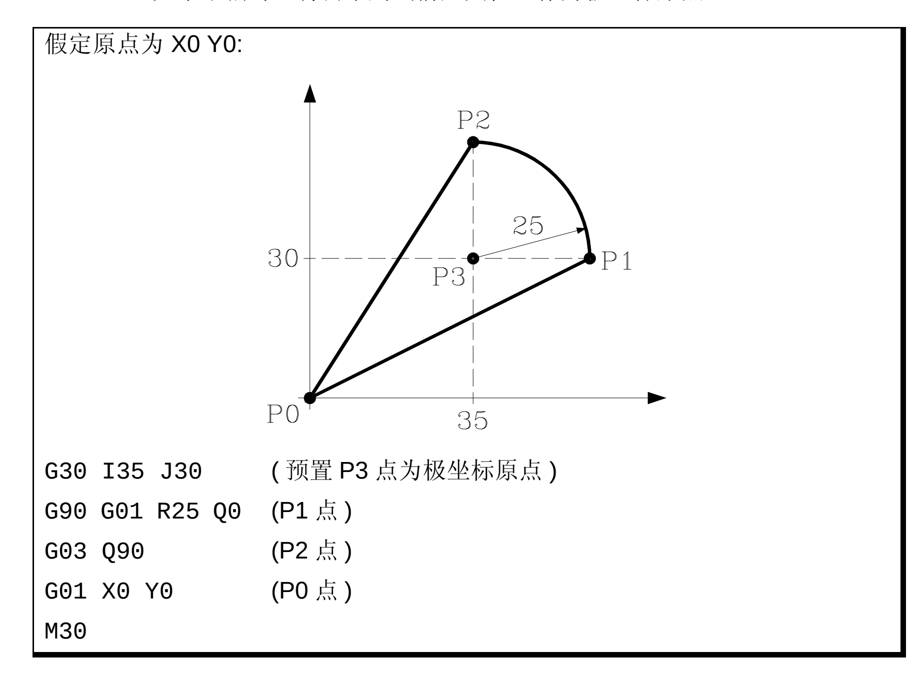
4.6 极坐标原点预置 (G30)

功能 G30 可以用于预置工作平面的任意点作为新的极坐标原点。如果不选择点，则采用当前参考系原点 （工件零点）作为极坐标原点。

编程

极坐标原点预置必须单段编写。编程格式是 "G30 I J"，其中：

I, J 它们是定义为新极坐标原点的点的横坐标和纵坐标。它们必须在绝对坐标方式下相对于工件零点被定义。
当编程时，两个参数都必须编写。
如果不编写，将会采用当前刀具位置作为极坐标原点。




```
G30 I35 J30      ( 预置 P3 点为极坐标原点 )
G90 G01 R25 Q0   (P1 点 )
G03 Q90          (P2 点 )
G01 X0 Y0        (P0 点 )
M30
```

因此，功能 G30 可以如下编写：

G30 I J　假定相对于工件零点的横坐标是 "I" 、纵坐标是 "J" 的点作为新的极坐标原点。

G30　当前刀具位置作为新的极坐标原点。

功能特性

功能 G30 是模态指令，极坐标原点保持有效，直到预置另一个值或改变工作平面。当工作平面变换时，采用那个平面的工件原点作为新的极坐标原点。

上电，编写指令 M02 或 M30 以及急停或复位之后，数控系统采用当前选择的工件零点作为极坐标原点。









工艺功能

5

5.1 加工进给率 (F)

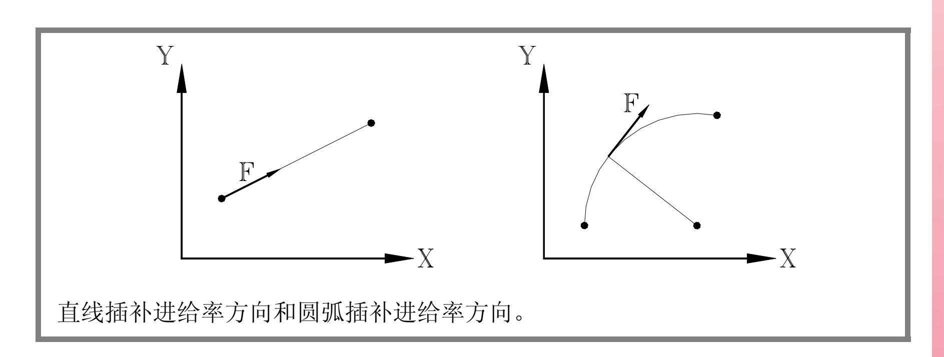
加工进给率可以通过编写 F 代码来选择，它保持有效直到编写另一个进给率值。编程单位根据现当前工作模式 (G93, G94 或 G95) 和运动轴的类型（直线型或旋转型）确定。

可以使用参数或算术表达式编程。

操作

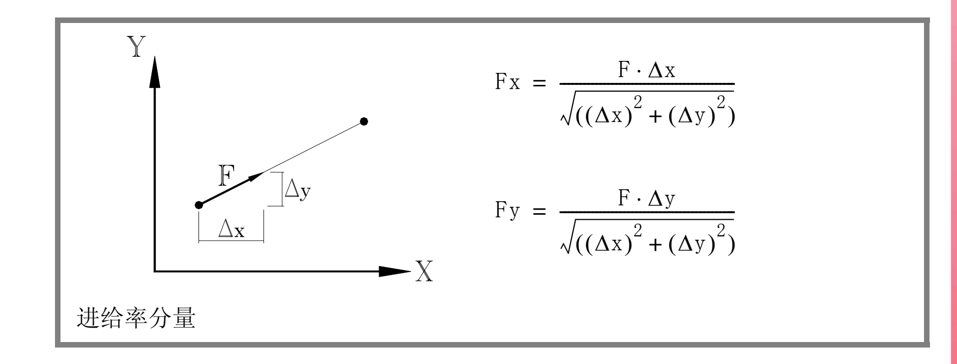
编写的进给率 F 对直线插补 (G01) 或圆弧插补 (G02, G03) 运动有效。 G00 （快速进给）方式运动执行机床参数 G00FEED 指定的快速进给率，与编写的 F 值无关。

进给率沿路径方向衡量，也可以是沿直线 （直线插补）方向或是圆弧的切线 （圆弧插补）方向。



直线插补进给率方向和圆弧插补进给率方向。

当插补仅包括主要轴 (X-Y-Z) 时，轴分量进给率和编写的进给率 F 之间的关系就像轴的运动和编写的运动合成之间的关系一样。



$$Fx = \frac{F \cdot \Delta x}{\sqrt{((\Delta x)^2 + (\Delta y)^2)}}$$

$$Fy = \frac{F \cdot \Delta y}{\sqrt{((\Delta x)^2 + (\Delta y)^2)}}$$

进给率分量

当插补包括旋转轴时，计算旋转轴的进给率，所以旋转轴运动的开始和结束与主轴运动的开始和结束相一致。如果计算出的旋转轴的进给率大于允许的最高极限速度，数控系统将改变 F 值使旋转轴可以以其最大速度旋转。



进给率倍率

编程的进给率 F 可以通过数控操作面板上的选择开关在 0% 到 200% 之间进行倍率修调，也可以通过编程或 PLC 进行选择。然而，最大修调率受机床参数 MAXOVR 值限制。

如果工作在快速定位方式 G00，快速进给率倍率修调率将固定在 100% 。也可以在 0% 到 100% 之间变化，这取决于参数 RAPIDOVR 的设置。

当执行螺纹加工时，进给率倍率设置为编程进给率的 100% 。

5.2 进给率相关功能

5.2.1 进给率编程单位 (G93/G94/G95)

与编程单位相关的功能将决定选择使用毫米 / 分 (英寸 / 分) 还是毫米 / 转 (英寸 / 转)，或者说是轴运动到目标位置所需要的时间。

编程

与编程单位相关的功能是：

G94 进给率单位：毫米 / 分 (英寸 / 分)。

G95 进给率单位：毫米 / 转 (英寸 / 转)。

G93 加工时间单位：秒

这些功能可以编写在程序的任何位置，并且不必单段编写。

如果运动轴是旋转轴，编程单位中是度而不是毫米或英寸，如下：

	直线轴	旋转轴
G94	毫米 (英寸)/ 分	度 / 分
G95	毫米 (英寸)/ 转	度 / 转
G93	分	分

G94 **进给率单位：毫米 / 分 (英寸 / 分)**

执行 G94 后，数控系统认为 F 代码编写的进给率单位为毫米 / 分 (英寸 / 分)。如果运动轴是旋转轴，则数控系统认为编程进给率的单位是度 / 分。

G95 **进给率单位：毫米 / 转 (英寸 / 转)**

执行 G95 后，数控系统认为 F 代码编写的进给率单位为毫米 / 转 (英寸 / 转)，每转指是主轴每转一转。如果运动轴是旋转轴，则数控系统认为编程进给率的单位是度 / 转。

此功能不影响 G00 方式下的运动， G00 总是在毫米 / 分 (英寸 / 分) 单位下执行。

G93 **加工时间单位：秒**

执行 G93 后，数控系统认为运动必须按 F 代码指定的时间完成，单位为秒。

此功能不影响 G00 方式下的运动， G00 总是在毫米 / 分 (英寸 / 分) 单位下执行。



功能特性

功能 G93， G94 和 G95 是模态指令，互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G94 还是 G95 取决于对机床参数 IFEED 的设置。



5.2.2 进给率混合 (G108/G109/G193)

用这些功能可以混合使用不同进给率编写连续程序段之间的进给率。

编程

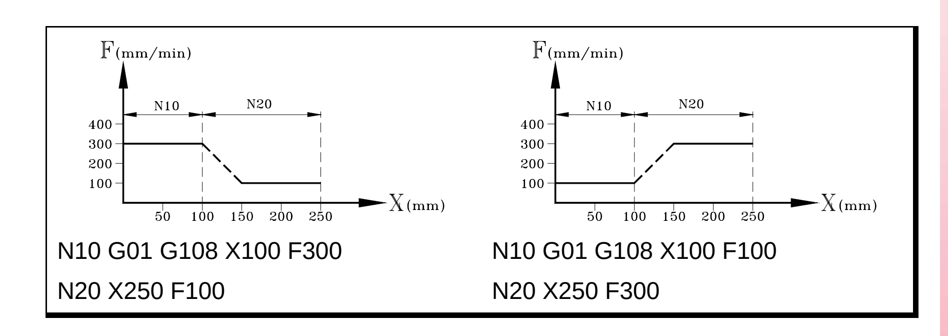
进给率混合相关的功能：

G108	在程序段开端混合进给率。
G109	在程序段末尾混合进给率。
G193	插入进给率。

这些功能可以编写在程序的任何位置，不必单段编写。

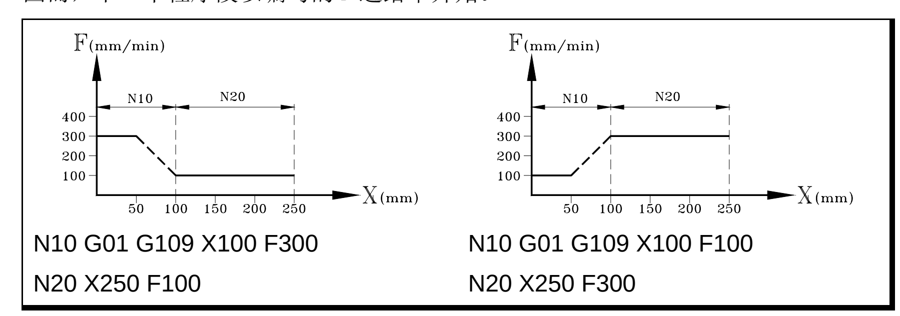
G108 在程序段开始混合进给率

当 G108 起作用时，针对新进给率的调速 （加速或减速）在下一个程序段开始时发生。当前程序段以 F 编写的进给率结束 。



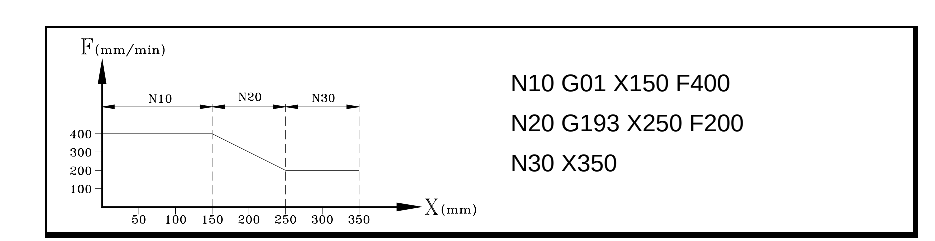
G109 在程序段末尾混合进给率

当编写 G109 时，针对新进给率的调速（加速或减速）在当前程序段的末尾时发生，因而，下一个程序段以编写的 F 进给率开始。



G193 进给率插入

当编写 G193 时，针对新进给率的调速线性地插入在程序段中编写的运动之间。



注意事项

尽管缺省功能是 G108 (在程序段开始混合进给率) ；在从 G00 转换到 G01、 G02 或 G03 期间，进给率混合总是发生在编写 G00 的程序段的末尾。

仅当厂商通过参数 SLOPETYPE 设置机床以线性加速度工作时，才应用进给率插入。其余情形下，进给率在程序段的开始进行调整 (G108)。

仅当厂商设置机床以梯形或方正弦形 （钟形）加速度运转时，应用功能 G109。

功能特性

功能 G109 和 G193 是非模态的， G108 是模态的，彼此不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G108。



5.2.3 恒进给率方式 (G197/G196)

用此项功能，可以选择加工时刀心进给率或者切削点的进给率恒定不变，以此保证在刀具半径补偿方式下工作时，编写的F值和刀具与工件间的接触点的进给率一致。

编程

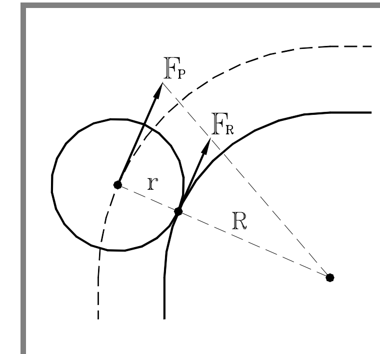
与进给率类型相关的功能是：

G197	恒刀心进给率。
G196	恒切削点进给率。

这些功能可以编写程序的任何位置，不必单段编写。

G197 **恒刀心进给率**

执行 G197 后，数控系统认为编写的 F 值与刀心进给率一致。意味着切削点进给率在内弧时增大，在外弧时减小。



恒切削点进给率：

$$F_R = \frac{R}{R+r} \cdot F_P$$

位置：

F_P	编程进给率
R	路径半径
r	刀具半径

G196 **恒切削点进给率**

执行 G196 后，数控系统认为编写的 F 值和刀具与工件的接触点的进给率一致。可得到一致的工件表面，甚至在圆弧处。

适用恒进给率的最小半径

使用指令 "#TANGFEED RMIN [<radius>]"，可以设置最小半径，恒切线进给率仅适用于圆弧半径大于最小设置时。如果没有编写或设置为零，数控系统在所有圆弧处采用恒切线进给率。

最小半径从下一个运动程序段应用，其值保持到执行 G197 后。

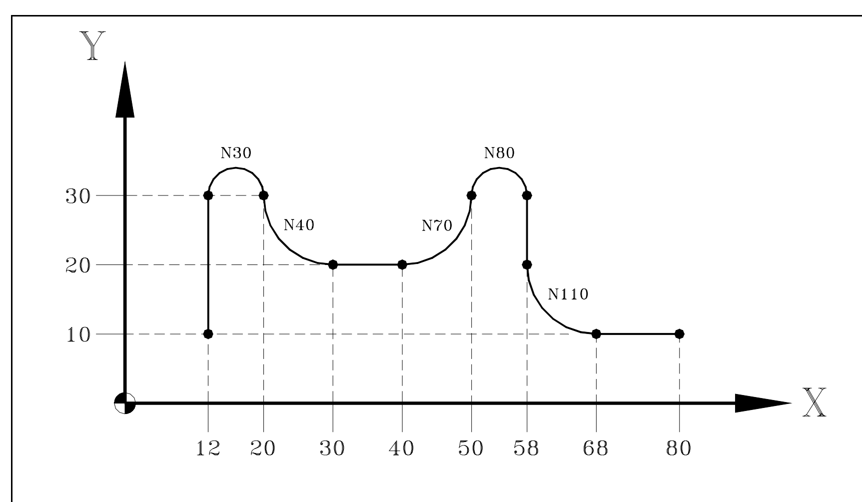




功能特性

功能 G197 和 G196 是模态的，互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G197。



N10 G01 G196 G41 X12 Y10 F600	(刀具补偿，恒切线进给率)
N20 G01 X12 Y30	
N30 G02 X20 Y30 R4	(恒切线进给率)
N40 G03 X30 Y20 R10	(恒切线进给率)
N50 #TANGFEED RMIN [5]	(最小半径 = 5)
N60 G01 X40 Y20	
N70 G03 X50 Y30 R10	(恒切线进给率)
N80 G02 X58 Y30 R4	(非恒切线进给率。 $R_{PROGRAMMED} < R_{MINIMUM}$)
N90 G01 X58 Y20	
N100 #TANGFEED RMIN [15]	(最小半径 = 15)
N110 G03 X68 Y10 R10	(非恒切线进给率。 $R_{PROGRAMMED} < R_{MINIMUM}$)
N120 G01 X80 Y10	
N130 G01 G40 X100	
N140 M30	



5.2.4 进给率修调 % 的取消 (G266)

G266 **进给率修调 100%**

该功能设置进给修调在 100%，不能通过面板上的选择键或 PLC 改变该设置。

功能 G266 仅影响它所在的程序段，因此，在定义运动的程序段添加它才有意义 。



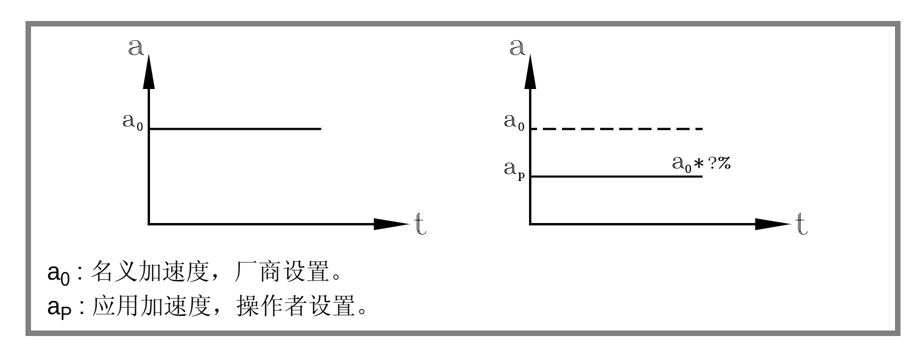
5.2.5 加速度控制 (G130/G131)

此项功能允许改变轴的加速度和减速度。

编程

与加速度控制相关的功能：

G130	应用于每根轴的加速度百分率
G131	应用于所有轴的 100% 加速度



a_0 : 名义加速度，厂商设置。
a_p : 应用加速度，操作者设置。

G130 应用于每根轴的加速度百分率

应用于每根轴的加速度百分率设置方法是用 G130 后跟轴和应用于每根轴的百分率。

应用的加速度值必须是整数 （不能是小数）。

```
...
G00 X0 Y0
G01 X100 Y100 F600
G130 X50 Y20            (X 轴加速度 = 50%)
                        (Y 轴加速度 = 20%)
G01 X0
G01 Y0
G131 100 X50 Y80        ( 对所有的轴恢复 100% 的加速度 )
                        ( 到点 X=50 Y=80 的运动 )
...
```

G131 应用于所有轴的 100% 加速度

应用于所有轴的 100% 加速度设置方法是用 G131 后跟新的应用于所有轴的加速度值。

应用的加速度值必须是整数 （不能是小数）。

当增加一个运动程序段时，在执行此运动前采用新的值。

注意事项

指令 #SLOPE 决定对如下情况中定义的值的影响：

- 快速点定位 (G00)
- 加速或减速阶段
- 加速或减速阶段的加加速度

编写的百分率是绝对值，也就是说，编写两次50%的意思是应用50%，而不是25%。

功能特性

功能 G130 和 G131 是模态的，互不兼容 。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统对所有轴采用 100% 加速度。



5.2.6 加加速度控制 (G132/G133)

可以通过此功能修改加加速度。

编程

与加加速度控制相关的功能：

G132	应用于每根轴的加加速度百分率
G133	应用于所有轴的 100% 加加速度

G132 应用于每根轴的加加速度的百分率

应用于每根轴的加加速度百分率设置方法是 G132 后跟轴和应用于每根轴的新的加加速度的百分率。

应用的加加速度值必须是整数 （不能是小数）。

```
G00 X0 Y0
G01 X100 Y100 F600
G132 X20 Y50        (X 轴加加速度百分率 = 20%)
                    (Y 轴加加速度百分率 = 50%)
G01 X0
G01 Y0
G133,100 X50 Y80    ( 恢复所有轴的 100% 加加速度。运动到点 X=50 Y=80)
```

G133 应用于所有轴的 100% 加加速度

应用于所有轴的 100% 加加速度设置方法是 G133 后跟新的应用于所有轴的加加速度值

应用的加加速度值必须是整数 （不能是小数）。

应用的加速度值必须是整数 （不能是小数）。

当增加一个运动程序段时，在执行此运动前采用新的加加速度值。

注意事项

指令 #SLOPE 决定是应用新的百分率，还是不应用于快速定位运动 (G00)。

编写的百分率是绝对值，换句话说，编写两次 50% 意味着应用 50%，而不是 25%。

功能特性

功能 G132 和 G133 是模态的，互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统对所有轴采用 100% 加加速度。





5.2.7 前馈控制 (G134)

前馈控制可以用于减少跟踪误差量 (轴运动滞后)

可以通过机床参数、 PLC 和编程来应用前馈。 PLC 定义的值有最高优先权，机床参数定义的值优先权最低。

编程

G134 应用的前馈百分率

前馈百分率设置方法是由 G134 后跟轴和新的应用于每根轴的前馈百分比。

前馈值可以最多可定义两位小数。

G134 X50.75 Y80 Z10	(应用的前馈值百分率：)
	(应用于 X 轴 = 50.75%)
	(应用于 Y 轴 = 80%)
	(应用于 Z 轴 = 10%)

注意事项

应用的前馈值的最大值是 120%。

编写的百分率是绝对值，换句话说，编写两次 50% 意味着应用 50%，而不是 25%。

G134 定义的值优先于机床参数定义的值，但是优先权低于 PLC 定义的值。

功能特性

功能 G134 是模态的。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用机床厂商为每根轴设置的前馈。

用于从 PLC 定义前馈的变量

变量 (V.)A.PLCFFGAIN.Xn 可以用于从 PLC 为每根轴设置前馈。用这个变量定义的值优先于机床参数或编程定义的值。

为此参数设置负值将取消其作用 (零值也是有效的)。不能在复位或确认参数时初始化该变量。



5.2.8 交流前馈控制 (G135)

交流前馈控制可以用于改进在改变加速度时的系统响应，以及减少加速度和减速度阶段的跟踪误差 （轴的延迟）量。

通过机床参数、 PLC 和编程可以应用交流前馈 。PLC 定义的值具有最高优先权，机床参数定义的值优先级最低。

编程

G135 应用的交流前馈的百分率

交流前馈百分率设置用指令 G135 后加轴和应用在每根轴的新百分率。

应用的交流前馈值最多可以保留一位小数。

G135 X55.8 Y75 Z110	(应用的交流前馈百分率)
	(X 轴 = 55.8%)
	(Y 轴 = 75%)
	(Z 轴 = 110%)

注意事项

可应用的最大 交流前馈百分率是 120%。

百分率的编写是绝对方式，换句话说，编写两次 50% 意思是应用 50%，不是 25%。

用 G135 编写的值优先于机床参数编写的值，但是低于 PLC。

功能特性

功能 G135 是模态的。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统恢复机床参数给每根轴设置的交流前馈。

从 PLC 定义交流前馈的变量

可写变量 (V.)A.PLCACGAIN.Xn 可以用于通过 PLC 为每根轴设置交流前馈。变量定义的值优先于通过机床参数或编程定义的值 。

为此参数设置负值将取消其作用 (零值也是有效的)。不能在复位或确认参数时初始化该变量。



5.3 主轴转速 (S)

通过编写主轴名后跟预定转速选择主轴转速。所有主轴的转速可以编写在同一程序段中。

```
S1000
S1=500
S1100 S1=2000 S4=2345
```

编写的转速一直有效，直到编写其它转速。编程单位是 RPM，除非选择别的方式。

可以使用参数或算数表达式。

主轴开始和停止

定义转速并不意味启动主轴运动。启动主轴用下列辅助功能定义。 See *" 主轴控制 (M03/M04/M05/M19)"* on page 75.

M03 - 主轴顺时针旋转启动。

M04 - 主轴逆时针旋转启动。

M05 - 主轴停止。

最大转速

每个范围 （档）的最大转速由机床厂商限制。当编写高转速时，数控系统限制其值在当前范围 （档）允许的最大值以下。通过操作面板的 "+" 和 "-" 按键或 PLC 以及编程方式试图超越最大极限时，也同样如此。

转速倍率

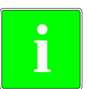
编写的 "S" 转速可以在 50% 到 120% 之间进行倍率修调，使用操作面板的 "+" 和 "-" 键或 PLC 实现。然而，最大和最小的转速倍率可以是不同的，这取决于机床厂商对机床参数 MINOVR 和 MAXOVR 的设置。

同样地，使用操作面板上的 "+" 和 "-" 键来改变编写的主轴转速 "S" 时，相关的增量步长是10；但是，这个值可以是不同的，这取决于轴的机床参数STEPOVR的设置。

在螺纹加工期间，编写的转速不能进行倍率修调，它设置为程序编写的 "S" 转速值的 100%。



5.3.1 主轴转速编程



以下功能针对于车床。为了使恒表面速度模式可用，机床必须设置其中一根轴 - 曲轴 - (通常是垂直于零件轴的轴)。

跟主轴转速编程有关的功能可以用于选择恒表面速度模式还是恒旋转速度模式。恒表面速度仅对与通道连接的主轴可用。

在恒表面速度方式时，当垂直轴运动时数控系统改变主轴转速，是为了保持刀具和工件之间切削速度不变，从而优化切削条件。

编程

与主轴转速编程相关的功能如下：

G96 恒表面速度。

G97 恒旋转速度。

这些功能可以编写在程序的任何地方，不必单段书写。

G96 **恒表面速度**

功能 G96 仅影响该通道连接的主轴。

在执行 G96 之后，数控系统认为该通道连接的主轴编写的主轴转速单位是米 / 分 (英尺 / 分)。在 G96 起作用时，这种工作方式在编写新速度时有效。

推荐在功能 G96 所在程序段中编写速度。主轴档位（范围）(M41, M42, M43, M44) 必须在同一程序段或在前面的程序段中选择。

G97 **恒旋转速度**

功能 G97 影响该通道连接的全部主轴。

执行功能G97后，数控系统认为编写的主轴转速单位是rpm，在恒转速下开始工作。

推荐在功能 G97 所在程序段编写速度；如果不编写，数控系统采用主轴当前转速作为编程速度。主轴档位 （范围） (M41, M42, M43, M44) 可以在任何时候选择。

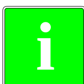
功能特性

功能 G96 和 G97 是模态指令，互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G97。

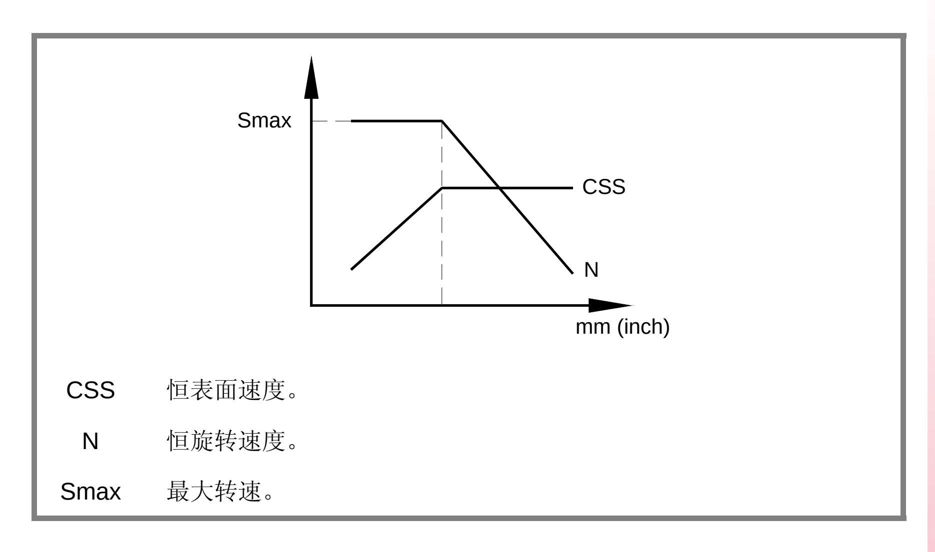


5.3.2 转速极限



以下功能针对车床。

当工作在恒表面速度时，由于转速随垂直轴的运动而变化时，因此必须编写最大转速。当主轴达到最大转速时，保持工作在恒转速方式。



CSS 恒表面速度。

N 恒旋转速度。

Smax 最大转速。

此极限仅对与通道连接且工作在恒切削速度时的主轴有效。当工作在恒转速并且最大许可转速为当前档位 （范围）设置的一个时，可以忽略它。

G192 在恒切削速度模式下的转速极限

转速极限通过功能 G192 设置， 恒表面速度的最大转速也是。最大转速的单位始终是 RPM。

当执行 G192 时，数控系统限制 "S" 设置的最大转速 。主轴转速不能超过 G96 的值，更不能编写更高的转速。最大转速也不能通过操作面板上的 "+" 和 "-" 键来超越。

G192 S2500	最大转速 = 2500 rpm
G96 S180	恒表面速度 =180m/min.
...	
G97 S1000 M3	恒转速 = 1000RPM
...	
G96	
...	
S230	激活恒表面速度模式。 转速极限保持作用在 2500RPM.



5.4 刀号 (T)

利用 T 代码可以选择刀具。刀具可以是在数控系统管理的刀库或手工的刀具表 (也称基础刀库) 里。

编程格式 T<0-4294967294>，并且可以利用参数和算术表达式编程。在这些情况下，计算值四舍五入到一个整数。如果结果是负数，系统发布相关错误报告。

定义

在主轴上安装刀具，必须提前定义。为此，数控系统提供一个表格，用户可以在这个表里定义每一把刀具的数据。

另一方面，在由数控系统管理的刀库中，必须定义每把刀具占用的刀库位置。为此，数控系统提供一个表格，用户可以定义每把刀具的位置。

表格数据可以通过以下方式定义：

- 手动从数控系统的面板 (如操作手册所述)。
- 通过编程，用相关变量 (如本手册相关章节所述)。

在主轴上安装刀具

加工所需刀具可以通过编程时使用代码 "T<n>" 来选择，其中 <n> 是要安装在主轴上的刀具的编码。

"T" 代码仅选择刀具。选择刀具之后，必须编写功能 M06 安装刀具到主轴上。安装 / 卸载依靠关联 M06 的子程序完成 （假如机床厂商这样定义的话）。

```
N10 G00 X0 Y0 F500 S1000 M03
N20 T1                  ( 选择刀具 T1)
N30 M06                 ( 安装 T1 到主轴 )
N40 ...
N50 T2                  ( 选择 T2)
N60 ...
N70 ...
N80 ...
N90 M06                 ( 安装 T2 到主轴 )
N100 ...
N110 M30
```



在刀库装卸刀具

将刀具装入刀库，刀库必须在装载模式。将刀具从刀库卸下，刀库必须在卸载模式。刀具经过主轴从换刀位置被装入刀库，经过主轴被卸下到换刀位置。

刀库的工作模式由变量 V.[n].TM.MZMODE 设置，n 是通道号码。根据变量的值，管理器将采取以下的工作模式。

值	意义
0	正常模式 (缺省和复位后)。
1	刀库装载模式。
2	刀库卸载模式。

一旦刀库在装载或卸载模式下，将通过程序中编写的代码 T*n* *执行操作*，*n* 是刀号。一旦刀具安装或卸载完毕，刀库必须设置成正常模式 (·0· 值)。

```
V.[1].TM.MZMODE = 1
T1 M6
T2 M6
...
V.[1].TM.MZMODE = 0
```

在特定刀库位置装载刀具

由于其特殊性，某些刀具 (尺寸，重量等) 必须安装在特定的刀库位置 。

指令 POS*n* 为刀具定义刀库位置。它必须始终和 T*n* 编写在同一程序段中。

```
V.[1].TM.MZMODE = 1
T3 M6 POS24
    (Places tool 3 in magazine position 24)
...
V.[1].TM.MZMODE = 0
```

只有当刀库处于加载模式时才可以选择刀库位置。否则，发出相应错误报告。

在有多个刀库的系统装载刀具

当使用多个刀库时，必须指出刀具被装入哪个刀库，使用代码 MZ*n*， *n* 表示刀库编号。必须始终与 T*n* 编写在同一程序段中。

```
T1 MZ1 M6
    ( 放置 1 号刀具到第一刀库 )
T8 MZ2 POS17 M6
    ( 放置 8 号刀具到第二刀库 )
```




注意事项

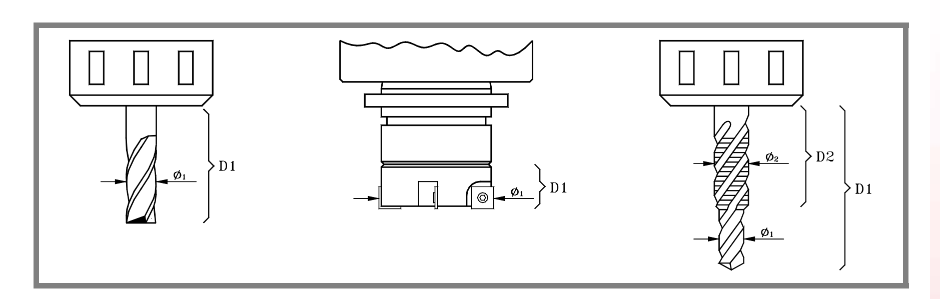
机床厂商可以关联一个子程序与 "T" 代码，当选择刀具时自动执行。如果 M06 包括在子程序里，当执行 "T" 代码时，刀具将被装到主轴上。





5.5 刀具偏置号 (D)

刀具偏置包含刀具尺寸。每把刀具可以有多个关联的偏置，当使用具有不同尺寸部件的组合刀具时，不同的部件可使用不同的偏置号。



当激活刀具偏置时，数控系统采用偏置定义的刀具尺寸，因此，当刀具半径或长度补偿有效时，数控系统会应用这些尺寸来补偿路径。

定义

要激活偏置，必须预先定义。为此，数控系统留出刀具表的一部分，在此用户可以定义多个偏置。偏置表数据可以以下方式定义：

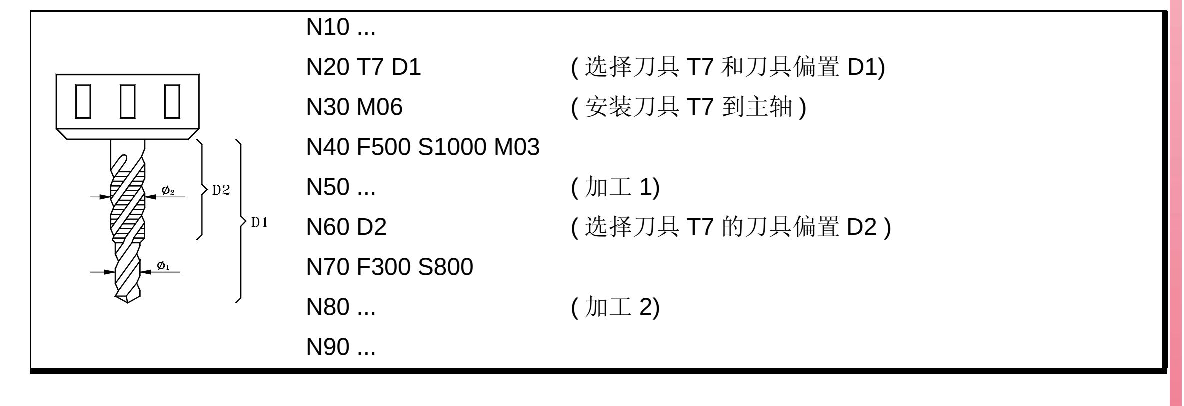
- 手工从数控系统面板输入 (如操作手册所述)。
- 通过编程，用相关变量 (如本手册相关章节所述)。

偏置仅和它们定义的刀具关联。即激活刀具偏置时，相应刀具的偏置被激活。

激活

一旦在表中定义刀具偏置，可以通过在程序中使用代码 "D<n>" 来选择， <n> 是应用的偏置号。偏置号也可以用参数或算数表达式来定义。

如果没有编写偏置，数控系统采用刀具偏置 D1。



```
N10 ...
N20 T7 D1            ( 选择刀具 T7 和刀具偏置 D1)
N30 M06              ( 安装刀具 T7 到主轴 )
N40 F500 S1000 M03
N50 ...              ( 加工 1)
N60 D2               ( 选择刀具 T7 的刀具偏置 D2 )
N70 F300 S800
N80 ...              ( 加工 2)
N90 ...
```

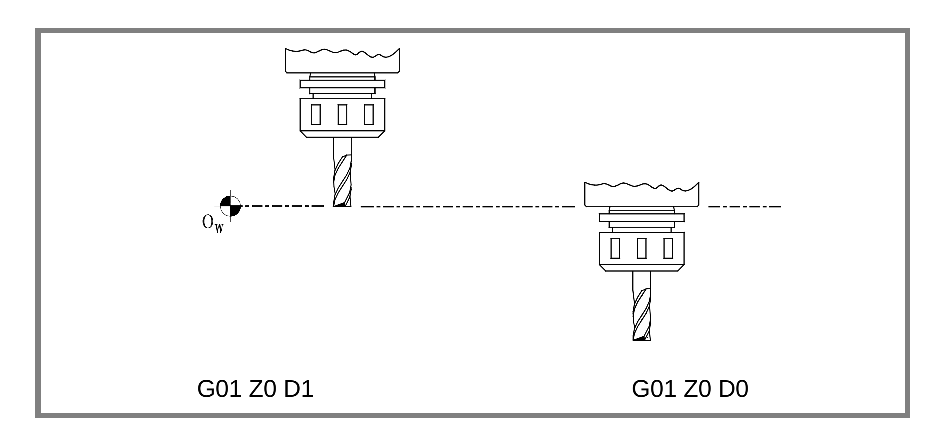
每次仅可以激活一个刀具偏置，所以，激活一个刀具偏置将取消前一个刀具偏置。编写 "D0" 将取消当前偏置。

N10 ...	
N20 T1 M06	(选择和安装刀具 T1。缺省，偏置 D1 激活)
N30 F500 S1000 M03	
N40 ...	(加工 1)
N50 T2	(准备刀具 T2)
N60 D2	(选择刀具 T1 的刀具偏置 D2)
N70 F300 S800	
N80 ...	(加工 2)
N90 M6	(安装刀具 T2 和它的偏置 D1)
N100 F800 S1200 M03	
N110 ...	(加工 3)
N120 ...	

注意事项

激活刀具偏置也激活刀具长度补偿。换刀后也激活此补偿，因为刀具变化后将采用相应刀具的 "D1"(如果没有编写其它的)。

用 "D0" 取消刀具补偿也取消刀具长度和半径补偿。





5.6 辅助功能 (M)

辅助功能 "M" 和所有数控程序执行以及各种机床装置的控制有关系，例如主轴换档，冷却液的开关，换刀等。

编程

在同一程序段内最多可以编写 7 个 M 功能。编程格式是 M<0 - 65535>，辅助功能可以用参数和算数表达式编程。在这些情况下 （缺省状态下），计算所得值四舍五入到整数。如果结果是负值，数控系统发出相关错误报告。

执行

依据机床厂商的设置 (M 功能表)：

- M 功能可以在编写的程序段的运动之前或之后执行。

  如果设置 M 功能在程序段运动之后执行，将取决于当前的功能 G05 或 G07：

  G05　　使用运动的理论终点执行 M 功能 (轴没有到达位置)。

  G07　　使用运动的实际终点执行 M 功能 (轴已经到达位置)。

- 数控系统等待或不等待确认，确认 M 功能在重新开始程序执行之前已经执行。如果不得不等待确认，在它编写的程序段的运动执行后必须接受确认。
- 表中没有设置的M功能在它们编写的程序段运动之前执行，并且在执行该程序段运动之前数控系统会等待 "M- 完成 " 的确认。

某些 M 功能被数控系统赋予特殊的内部含义。本章的 *"5.6.1 M 功能表 "* 部分提供了一份对于数控系统有内部含义的这些 M 功能的清单。

关联子程序

M 功能可以有一个相联的子程序，该子程序将代替功能执行。

如果在执行和 M 辅助功能相关的子程序时，程序段中含有相同的 M 功能，该功能将被执行，但不执行相关子程序。



5.6.1 M 功能表

程序中断 (M00/M01)

M00 程序停止

功能 M00 中断程序执行。不停止主轴或初始化切削条件。

再次启动程序，须按下操作面板的 [循环启动] 键 。

应该在 M 功能表中设置该功能，这样一来在编写该指令的程序段的末尾执行它。

M01 条件停止

当激活外部条件停止开关时 （来自 PLC 的 M01 STOP 信号），中断程序执行。不停止主轴或初始化切削条件。

启动程序执行，须再次按下操作面板的 [循环启动] 键 。

应该在 M 功能表中设置该功能，这样一来在编写该指令的程序段的末尾执行它。

程序结束 (M02/M30)

M02/M30 程序结束

两个功能都表示程序结束。执行此功能使该通道初始化并且选择程序的第一程序段。也停止主轴 (假如参数 SPDLSTOP 定义的话)，并且初始化切削条件。

应该在 M 功能表中设置该功能，这样一来在编写该指令的程序段的末尾执行它。

子程序结束 (M17/M29)

M17/M29 子程序结束

两个功能都表示子程序结束。



换刀 (M06)

M06 换刀

M06 功能执行换刀。数控系统控制刀具更换和为刀库更新列表。

应该在 M 功能表中设置该功能， 以便执行相应的换刀子程序。

主轴控制 (M03/M04/M05/M19)

M03 主轴顺时针旋转启动

M04 主轴逆时针旋转启动

功能 M03 启动主轴顺时针旋转，功能 M04 启动主轴逆时针旋转。功能保持有效，一直到编写另一主轴控制功能 (M03/M04/M05/M19)。

应该在 M 功能表中设置该功能， 这样一来在编写该指令的程序段的末尾执行它。

这些功能可以和编写的主轴转速在同一程序段或不同的程序段定义。如果编写功能的程序段没有提及任何主轴，该通道的主轴将执行它。

```
S1000 M3
    ( 主轴 S 以 1000 rpm 启动顺时针旋转 )
S1=500 M4
    ( 主轴 S1 以 500 rpm 启动顺时针旋转 )
M4
    ( 主轴启动逆时针旋转 )
```

如果一个程序段里编写了多根主轴，功能 M3 和 M4 应用于所有的这些轴 。要在不同的方向启动主轴，在 M 功能旁定义它所关联的主轴，如下所示。

M3.S / M4.S　　　M3 和 M4 关联主轴 S。

```
S1000 S2=456 M3
    ( 主轴 S 以 1000 rpm 旋转，主轴 S2 以 456 rpm 旋转，都是顺时针方向 )
M3.S S1000 S2=456 M4.S2
    ( 主轴 S 以 1000 rpm 转速顺时针旋转 )
    ( 主轴 S2 以 456 rpm 转速逆时针旋转 )
```

M05 主轴停止

功能 M05 停止主轴 。此功能保持有效直到编写另一主轴控制功能 (M04/M03/M19)。

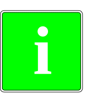
为停止一根主轴，在 M5 后定义和它相关的主轴，如下所示。如果没有提及任何主轴，应用于主要主轴。

M5.S　　　相关主轴 S 的功能 M5 。

```
S1000 S2=456 M5
    ( 停止主轴 )
M5.S M5.S2 S1=1000 M3.S1
    ( 停止主轴 S 和 S2)
    ( 主轴 S1 顺时针旋转 )
```

M19 主轴定位

功能 M19 使主轴定位。此功能保持有效直到编写速度控制功能。 (M03/M04/M05).



此工作模式仅在主轴上安装有旋转编码器的机床上可用。



当执行功能 M19 时，数控系统认为 Sn 代码引入的值指出主轴的角度位置。如果同一程序段编写多根主轴，功能 M19 应用于全部这些主轴。

角度位置用度为单位编程，并且总是用绝对坐标表示，因而不受功能 G90/G91 影响。

```
M19 S0
    ( 主轴 S 在 0º 定位 )
M19 S2=120
    ( 主轴 S2 在 120º 定位 )
M19 S1=10 S2=34
    ( 主轴 S1 在 10º 定位，主轴 S2 在 34º 定位 )
```

定位主轴到 0 度 位置，也可以在 M19 后编写定位的主轴来定义。

M19.S1   主轴 S1 定位在 0º。

```
M19.S4
    ( 主轴 S4 定位在 0º)
M19
    ( 主要主轴定位在 0º)
```

每个定位步骤需要一个 M19。没有 M19 的 S 代码被作为主轴的新转速，在下次使用功能 M03/M04 设定的主轴转速模式时应用。

定位的执行

当第一次执行 M19 功能时，主轴复位。然后编写的 M19 功能仅用于主轴定位。为主轴再次复位，可用功能 G74。

当执行功能 M19，执行定位动作步骤如下。

1. 主轴停止 (如果主轴是转动着的)。
2. 数控系统不工作在速度模式，转换到定位模式。
3. 如果第一次执行 M19 ，数控系统复位主轴 (复位搜索)。
4. 定位主轴在0º位或由S代码定义的角度位置(如果编写了的话)。为此，计算编程值的模块 (在 0 到 360º 之间) ，并且主轴到达此位置。



为主轴定位设置主轴方向

如果执行功能 M19 时，已有 M3 或 M4 起作用，即使速度是零，该功能将设置主轴定位方向。

如果没有 M3 或 M4 起作用，旋转方向依据机床参数 SHORTESTWAY 的设置而定。

- 如果是 SHORTESTWAY 型主轴，通过最短路线定位。
- 如果不是 SHORTESTWAY 型主轴；缺省时，定位方向同前一主轴运动方向。也可以用 M19 和定位方向来定义，如下所示。

M19.POS 正方向定位

M19.NEG 反方向定位

要设置特殊的主轴旋转方向，必须如下编程。

```
M19.POS S120 S1=50
    ( 应用正方向定位到主轴 S 和 S1)
M19.NEG.S1 S1=100 S34.75
    ( 应用反方向定位到主轴 S1)
```

为 SHORTESTWAY 型主轴编写定位方向时，忽略编写的方向。

定位速度

主轴 Sn 的定位速度用指令 Sn.POS 来定义，如下所示：

Sn.POS 轴 Sn. 的定位速度

```
M19 S.POS=120 S1.POS=50
    ( 主轴 S 以 120 rpm ， 主轴 S2 以 50 rpm 定位 )
```

定位速度以单位 rpm 给出。

如果没有编写定位速度，数控系统采用机床参数 REFEED1 设置的值。

```
N10 G97 S2500 M03
    ( 主轴以 2500 RPM 旋转 )
N20 M19 S50
    ( 主轴定位控制。复位搜索，定位在 50º)
N30 M19 S150
    ( 定位在 150º)
N40 S1000
    ( 新的主轴转速。主轴保持在定位模式 )
N50 M19 S-100
    ( 定位在 -100º)
N60 M03
    ( 主轴转速控制。主轴以 1000 RPM 旋转 )
N70 M30
```

换档 (M41-M44)

M41-M44 主轴换档

编程转速要求的主轴档位 (速度范围) 用功能 M41, M42, M43 和 M44 选择。数控系统可以有最多 4 种不同的主轴档位 。

这些功能可以和编程的主轴定义在同一程序段或者在不同的程序段。如果在编写的程序段中没有提及任何主轴，将会应用它们到该通道的主要主轴。



```
S1000 M41
S1=500 M42
M44
```

使用 Sercos 轴时，功能 M41-M44 也包括变换驱动器的速度档。

如果在同一个程序段编写多根主轴，该功能将应用到所有的这些轴上。要对不同主轴应用不同的档，在 M 功能后定义相关的轴，如下所示：

M41.S 功能 M41 相关主轴 S。

```
S1000 S2=456 M41
    ( 主轴 S 和 S2 应用 1 档位 )
M41.S M42.S3
    ( 主轴 S 应用 1 档位 )
    ( 主轴 S3 应用 2 档位 )
```

每档最大转速由机床厂商限制。同样的，如果机床厂商设置了机床参数 AUTOGEAR，数控系统自动执行主轴换档，数控系统将根据编写的主轴转速自动运行 M41, M42, M43 和 M44 来 进行换档。



5.7 辅助功能 (H)

H 辅助功能用于发送信息给 PLC。不同于 M 功能的是，H 功能不等待批准便继续执行程序。

编程

在同一程序段最多可编写 7 个 H 功能。编程格式 H<0 - 65535>，可以用参数和算术表达式编程 。在这些情况下 （缺省状态下），计算值四舍五入到一个整数。如果结果是负值，数控系统将会发送错误报告。

执行

H 辅助功能在它所在程序段的开始被执行。





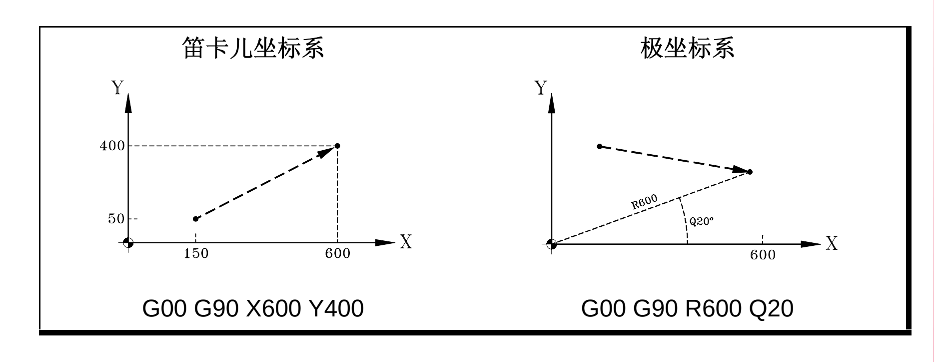


刀具路径控制

6

6.1 快速定位 (G00)

编写在 G00 后的运动为直线运动，以机床厂商设置的快速进给率从当前刀具位置运动到目标位置。无论包含几根轴，最终路径总是直线。



当快速定位包含辅助轴或旋转轴时，执行运动时，主轴开始或结束运动时它们开始或结束运动。

编程

运动可以定义如下：

- 在笛卡尔坐标系中 ("X","X1"..."C9")

  在各根轴定义终点坐标值。

  不是所有的轴都需要编程，只需编程运动轴。

- 在极坐标系中 ("R", "Q")

  终点的极坐标定义极半径和极角。

  极半径 "R" 是极坐标原点和该点之间的距离。极角 "Q" 是横坐标和极坐标原点与点之间的连线所成的角。

  如果没有编写极角或极半径，将保持为上一运动编写的值。



进给率运行方式

G00运动临时取消编写的F功能，快速定位运动执行机床厂商在参数G00FEED设置的值。当编写 G01, G02 或 G03 型的功能时， F 值恢复作用 。

当包含多根轴时，计算最终进给率，那么至少其中一根轴以最大速度运动。

在同一程序段定义 F 值和 G00 时 ，数控系统会存储赋给 F 的值，并在下次一时刻编写 G01, G02 或 G03 型功能时应用 。

倍率百分率设置在 100% ，或者根据机床厂商对参数 RAPIDOVR 的设置，可以通过操作面板上的按键在 0% 到 100% 之间修调。

功能特性

功能 G00 是模态的，和功能 G01, G02, G03, G33 以及 G63 互不兼容。

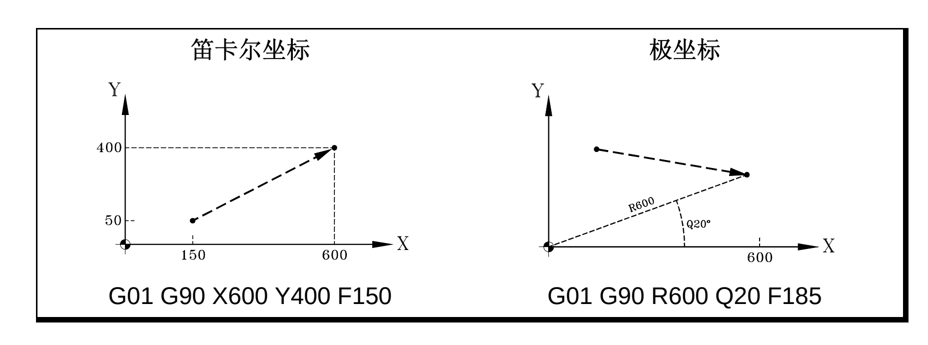
功能 G00 可以编写成 G0。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G00 还是 G01 取决于对机床参数 IMOVE 的设置。



6.2 直线插补 (G01)

编写在 G00 后的运动为直线运动，以编写的 F 进给率从当前位置移动到要求的目标点。无论包含几根轴，最终路径总是直线。



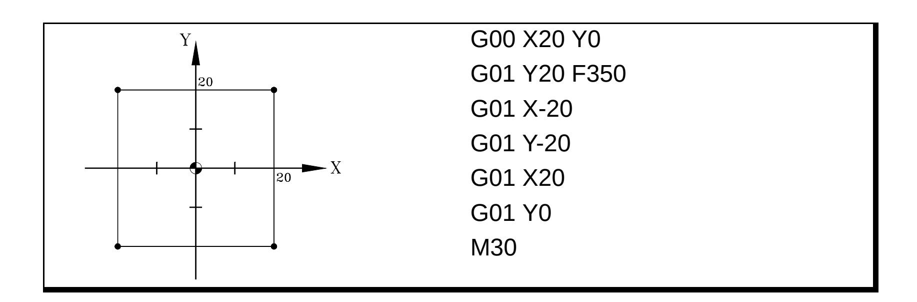
辅助轴和旋转轴也可以编写在直线插补程序段中。这时，数控系统计算这些轴的进给率，这些轴的运动开始和结束与主轴的同时进行。

编程

- 笛卡尔坐标系 ("X","X1"..."C9")

  在各根轴定义终点坐标值。

  不是所有的轴都需要编程，只需编写运动轴。

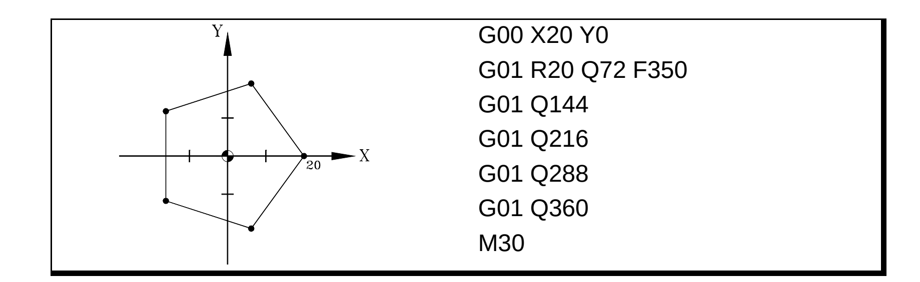


- 极坐标 ("R", "Q")

  终点的极坐标定义极半径和极角。

  极半径 "R" 是极坐标原点和该点之间的距离。极角 "Q" 是横坐标和极坐标原点与点之间的连线所成的角。

  如果没有编写极角或极半径，将会保持为最近运动所编写的坐标值。



进给率运行方式

编写的进给率 F 保持有效，直到编写新的值，因而不需要在每个程序段编写它。

当包含多根轴时，数控系统计算每根轴的进给率，所以最终路径以编写的进给率 F 执行。

编写的进给率 F 可以使用数控操作面板上的选择按键、程序或者 PLC，在 0% 到 200% 之间修调。然而，最大倍率由机床厂商通过参数 MAXOVR 限制。



辅助轴的进给率

辅助轴的运行方式由普通机床参数 FEEDND 决定。

- 如果它的值为真 （TRUE），没有轴会超过编写的进给率。
- 如果它的值为假（FALSE），进给率应用给主轴，然而辅助轴可以超越它，但是永远没有可以超越它们的 MAXFEED。如果有轴超越 MAXFEED，主轴的编程速度将被限制。

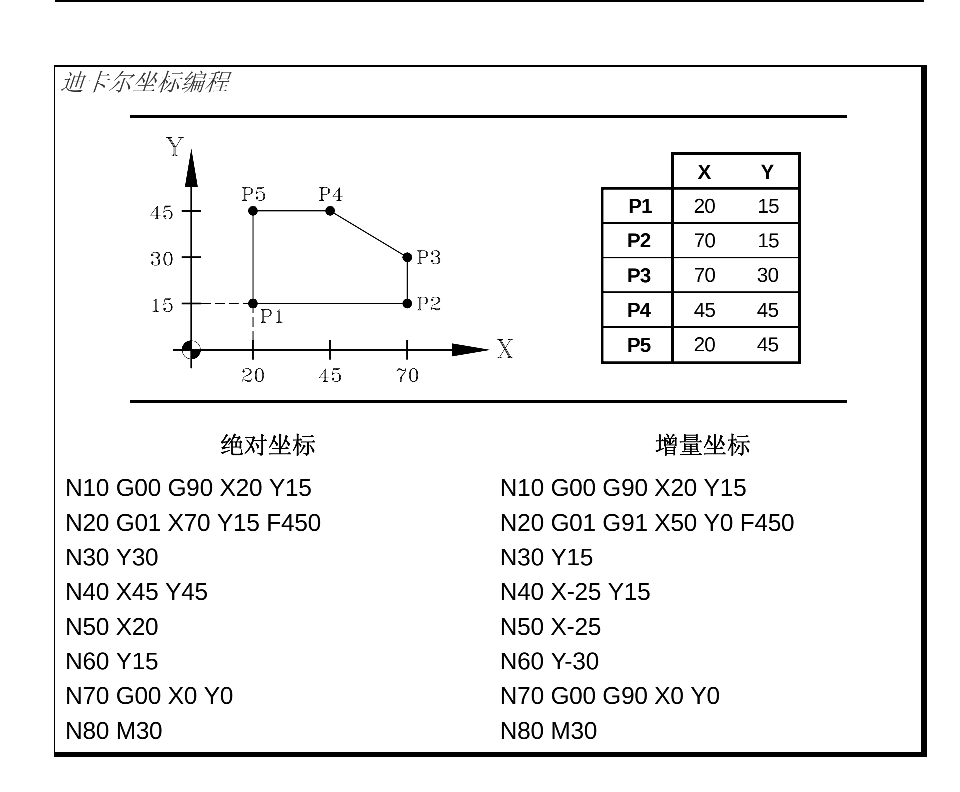
功能特性

功能 G01 是模态的，和 G00, G02, G03, G33 以及 G63 互不兼容。

功能 G01 可以编写成 G1。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G00 还是 G01 取决于对机床参数 IMOVE 的设置。

编程实例

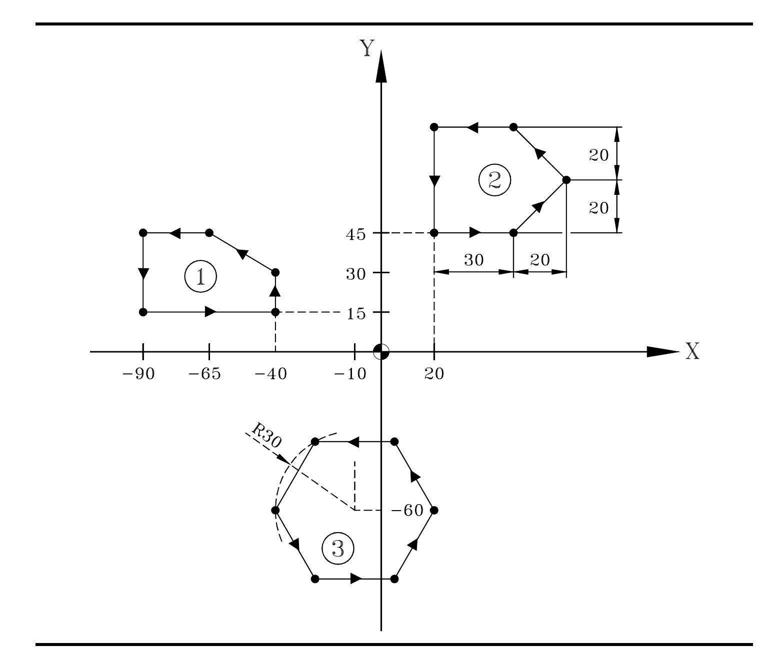


	X	Y
P1	20	15
P2	70	15
P3	70	30
P4	45	45
P5	20	45

绝对坐标	增量坐标
N10 G00 G90 X20 Y15	N10 G00 G90 X20 Y15
N20 G01 X70 Y15 F450	N20 G01 G91 X50 Y0 F450
N30 Y30	N30 Y15
N40 X45 Y45	N40 X-25 Y15
N50 X20	N50 X-25
N60 Y15	N60 Y-30
N70 G00 X0 Y0	N70 G00 G90 X0 Y0
N80 M30	N80 M30



笛卡尔坐标和极坐标编程




```
N10 T1 D1
N20 M06
N30 G71 G90 F450 S1500 M03        ( 初始条件 )
N40 G00 G90 X-40 Y15 Z10          ( 接近轮廓 1)
N50 G01 Z-5
N60 X-40 Y30                      ( 轮廓 1 加工 )
N70 X-65 Y45
N80 X-90
N90 Y15
N100 X-40                         ( 轮廓 1 加工结束 )
N110 Z10
N120 G00 X20 Y45 F300 S1200       ( 接近轮廓 2)
N130 G92 X0 Y0                    ( 预置新工件零点 )
N140 G01 Z-5
N150 G91 X30                      ( 轮廓 2 加工 )
N160 X20 Y20
N170 X-20 Y20
N180 X-30
N190 Y-40                         ( 轮廓 2 加工结束 )
N200 G90 Z10
N210 G92 X20 Y45                  ( 恢复先前的工件零点 )
N220 G30 I-10 J-60                ( 极坐标预制 )
N230 G00 R30 Q60 F350 S1200       ( 接近轮廓 3)
N240 G01 Z-5
N250 Q120                         ( 轮廓 3 加工 )
N260 Q180
N270 Q240
N280 Q300
N290 Q360
N300 Q60                          ( 轮廓 3 加工结束 )
N310 Z10
N320 G00 X0 Y0
N330 M30
```



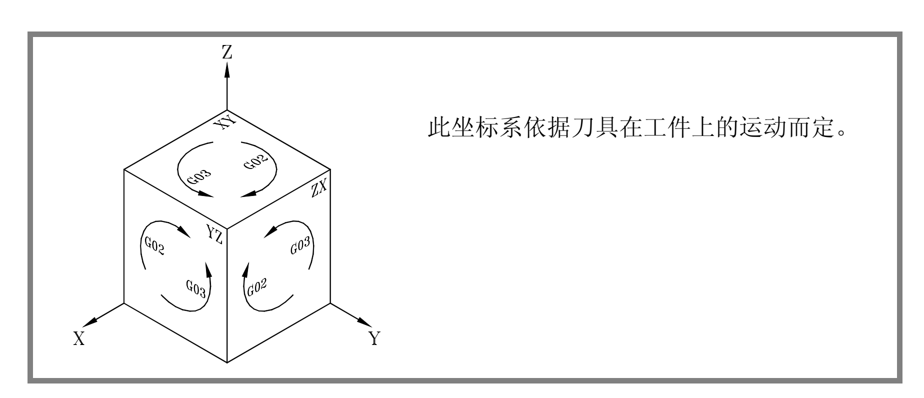

6.3 圆弧插补 (G02/G03)

编写在 G02 和 G03 后的运动执行圆弧路径，以编写的 F 进给率从当前位置移动到要求的目标点。

圆弧插补仅在当前平面执行，有两种圆弧插补 :

G02	顺时针圆弧插补。
G03	逆时针圆弧插补。

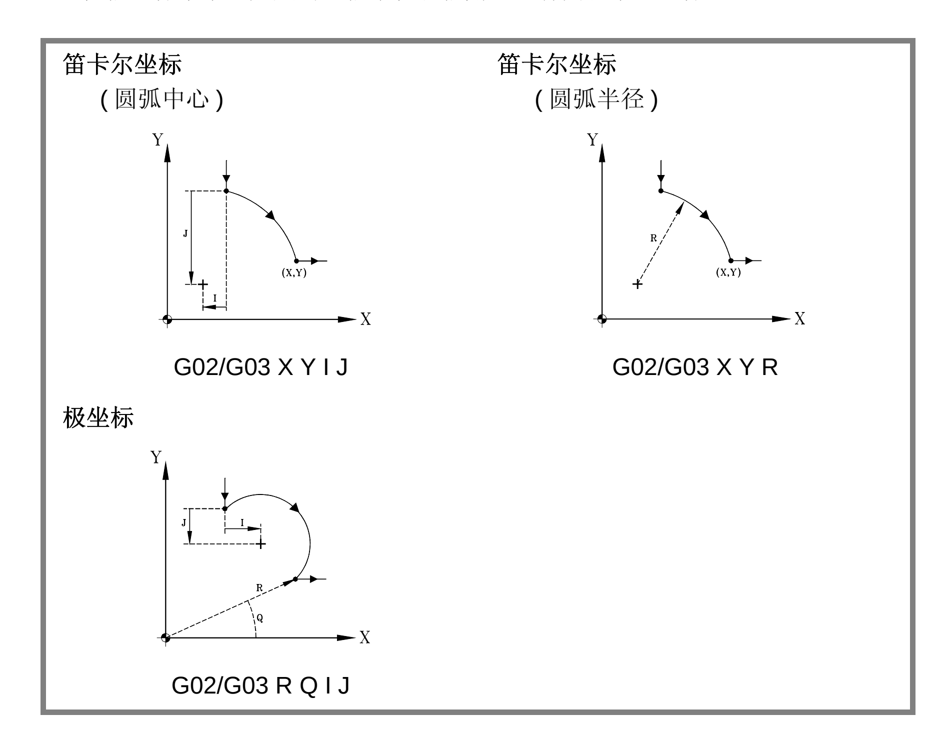
依照如下坐标系制定顺时针 (G02) 和逆时针 (G03) 运动方向。



编程

可以如下定义圆弧插补:

- 在笛卡尔坐标系中，定义终点坐标和圆心坐标。
- 在笛卡尔坐标系中，定义终点坐标和圆弧半径。
- 在极坐标系中，定义终点极半径和极角，还有圆弧中心坐标。



进给率运行方式

编写的进给率F一直有效，直到编写新的进给率值，因而不需要在每个程序段编写 。

编写的进给率 F 可以在 0% 到 200% 之间修调，通过数控系统操作面板上的选择按键、程序或 PLC 来选择。然而，最大倍率受到机床厂商对参数 MAXOVR 的设置的限制。

功能特性

功能 G02 和 G03 是模态的，彼此互不兼容，和 G00, G01, G33 以及 G63 互不兼容。

功能 G74 (复位搜索) 也可以取消功能 G02 和 G03。

功能 G02 和 G03 也可以编写成 G2 和 G3。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G00 还是 G01 取决于对机床参数 IMOVE 的设置。



6.3.1 笛卡尔坐标 (圆弧中心编程)

定义圆弧通过功能 G02 或 G03 后跟圆弧终点坐标和圆弧中心坐标 （相对于圆弧起点），依据当前工作平面的轴。

圆弧终点坐标

沿当前工作平面的轴定义坐标，可以使用绝对坐标或增量坐标。

如果没有编写或是和起点相同，将会执行一个整圆。

圆弧中心坐标

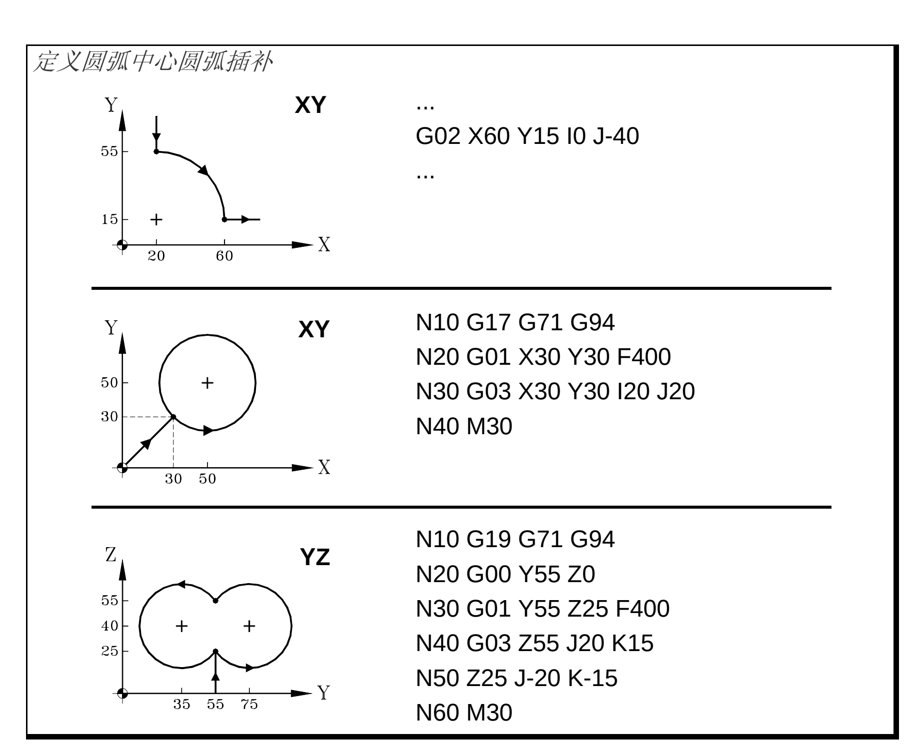
圆弧中心坐标用字母 "I", "J" 或 "K" 定义，这取决于当前工作平面。

G17 G18 G19	字母 "I", "J" 和 "K" 分别对应该通道的第一、第二和第三坐标轴。
G20	字母 "I", "J" 和 "K" 对应定义平面的横坐标轴、纵坐标轴和垂直轴。

当中心坐标在某根轴上为 0 时，不必编写。这些坐标不受功能 G90 和 G91 的影响。

根据当前平面，编程格式为：

XY 平面 (G17)	G02/G03	X...	Y...	I...	J...
ZX 平面 (G18)	G02/G03	X...	Z...	I...	K...
YZ 平面 (G19)	G02/G03	Y...	Z...	J...	K...



定义圆弧中心圆弧插补



6.3.2 笛卡尔坐标 (半径编程)

通过编写功能 G02 或 G03 后跟圆弧终点坐标和圆弧半径定义圆弧。

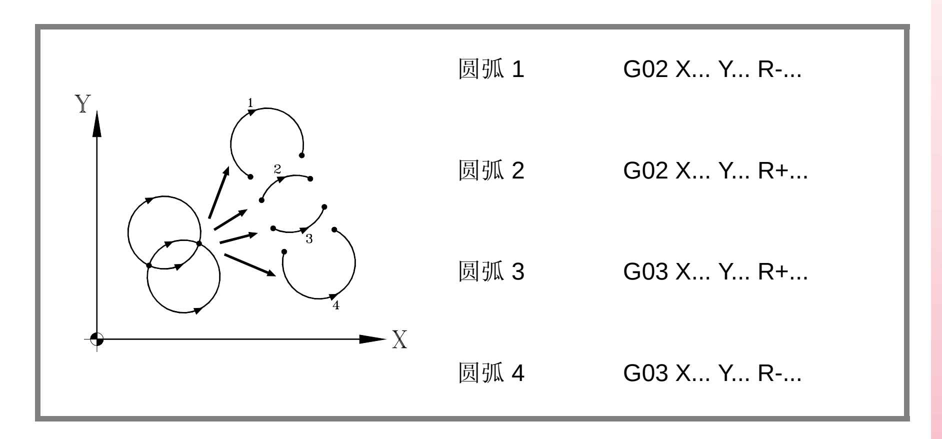
圆弧终点坐标

沿当前工作平面的轴定义坐标，可以用绝对坐标或增量坐标。

圆弧半径

圆弧半径定义用字母 "R" 或使用赋值 "R1=< 半径 >" or "G263=< 半径 >"。半径值一直有效，直到编写新的值或用中心坐标编写圆弧或用极坐标编写一个运动。

如果圆弧小于 180º，半径用正号编写。如果圆弧大于 180º，半径用负号编写。通过这种方法，并依据所选择的圆弧插补 (G02 或 G03)，便可得到预期的圆弧。



依据当前工作平面，编程格式为：

XY plane (G17)	G02/G03	X...	Y...	R+/-
ZX plane (G18)	G02/G03	X...	Z...	R+/-
YZ plane (G19)	G02/G03	Y...	Z...	R+/-

定义相同圆弧的不同格式。

XY	Nxx G03 G17 X20 Y45 R30 Nxx G03 G17 X20 Y45 G263=30 Nxx G03 G17 X20 Y45 R1=30
ZX	Nyy G03 G18 Z20 X40 R-30 Nyy G03 G18 Z20 X40 G263=-30 Nyy G03 G18 Z20 X40 R1=-30
YZ	Nzz G02 G19 Y80 Z30 R30 Nzz G02 G19 Y80 Z30 G263=30 Nzz G02 G19 Y80 Z30 R1=30

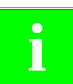
半径也可以编写在定义圆弧插补之前的程序段。此时，通过赋值 "R1=< 半径 >" 或 "G263=< 半径 >" 定义圆弧。

```
N10 G01 G90 X0 Y0 F500        N10 G01 G90 X0 Y0 F450
N20 G263=50                   N20 G01 G263=50
N30 G02 X100                  N30 G02 X100

N10 G01 G90 X0 Y0
N20 G02 G263=50
N30 X100
```

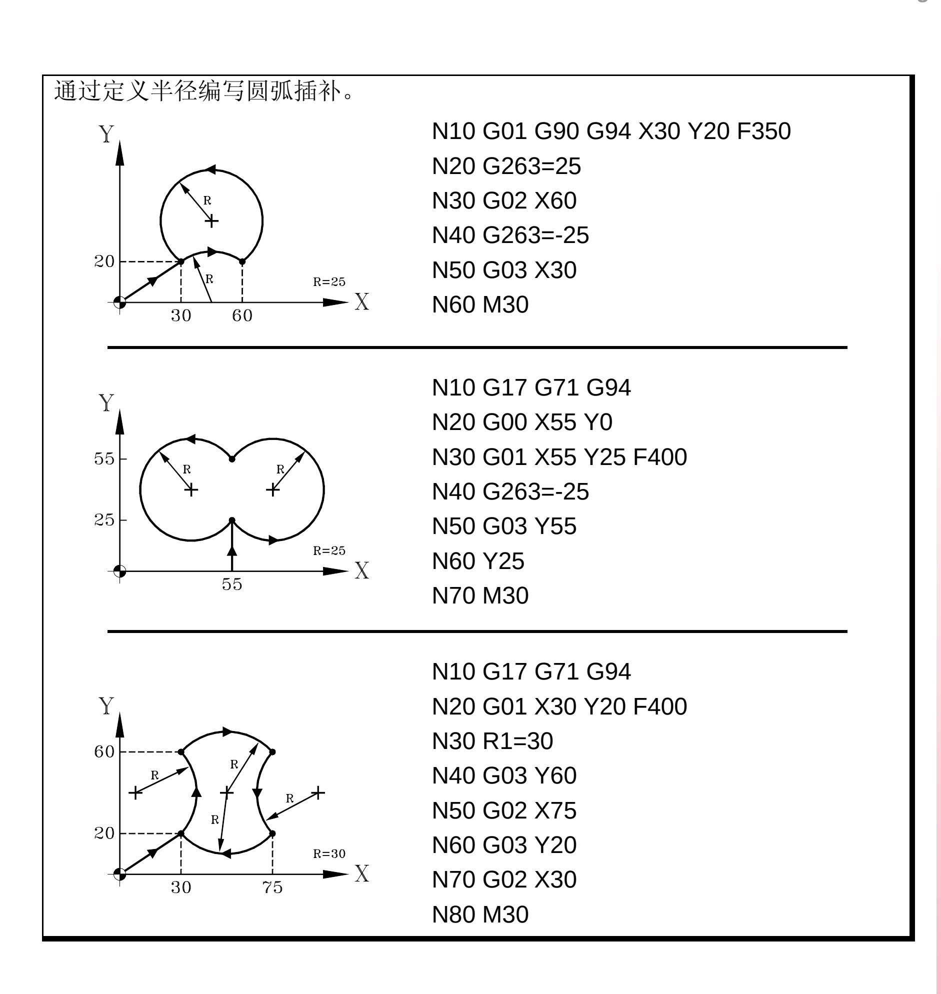
以上实例为加工半径是 50 mm 的半圆。虽然实例使用功能 "G263=<radius>"，如果编写 "R1=<radius>" 同样有效。

数控系统保存半径值，直到通过定义圆弧中心坐标编写一个圆弧插补或用极坐标编写一个运动。



使用半径编写圆弧时，不能编写整圆，因为那样将会有无数种可能。





通过定义半径编写圆弧插补。
N10 G01 G90 G94 X30 Y20 F350
N20 G263=25
N30 G02 X60
N40 G263=-25
N50 G03 X30
N60 M30
N10 G17 G71 G94
N20 G00 X55 Y0
N30 G01 X55 Y25 F400
N40 G263=-25
N50 G03 Y55
N60 Y25
N70 M30
N10 G17 G71 G94
N20 G01 X30 Y20 F400
N30 R1=30
N40 G03 Y60
N50 G02 X75
N60 G03 Y20
N70 G02 X30
N80 M30

6.3.3 极坐标

通过编写功能 G02 或 G03 后跟圆弧终点坐标和圆弧中心坐标（相对圆弧起点）定义圆弧，依照当前工作平面的坐标轴。

终点坐标

通过定义极半径 "R" 和极角 "Q" 给出终点坐标，如下所示：

Radius	极坐标原点和点之间的距离。
Angle	通过极坐标原点的水平坐标和极坐标原点与点之间的连线所成的角。

如果没有编写极角或极半径，将会保持为最近的运动所编写的坐标值。极半径和极角可以用绝对坐标 (G90) 或增量坐标 (G91) 定义。

当在 G91 方式下编写极角时，增量是相对于前一点和极坐标原点的连线；如果在 G90 方式下编写极角时，表示为和通过极坐标原点的水平坐标轴所成的角。

在 G91 方式下编写 360º 的角意味编写一个整圆。在 G90 方式下编写 360º 的角意味编写一个圆弧，终点和通过极坐标原点的水平坐标轴形成 360º 的角。

圆弧中心坐标

圆弧中心坐标依据当前工作平面用字母 "I", "J" 或 "K" 定义。

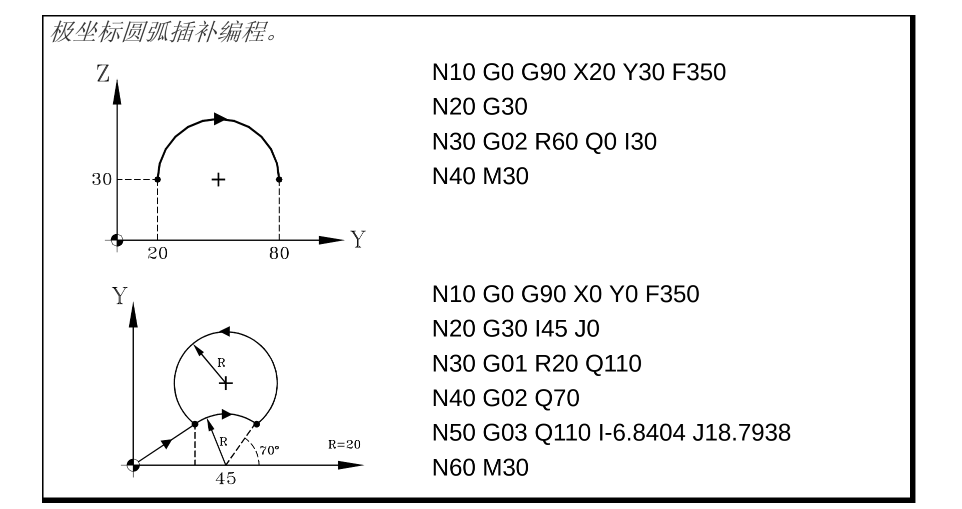
G17 G18 G19	字母 "I", "J" 和 "K" 分别对应该通道的第一坐标、第二坐标和第三坐标 。
G20	字母 "I", "J" 和 "K" 对应已定义平面的横坐标轴、纵坐标轴和垂直坐标轴。

当圆弧中心坐标在某根轴上为零时，不必编写；如果没有编写任何坐标值，将采用极坐标原点作为圆弧中心。圆弧中心坐标不受功能 G90 和 G91 的影响。

根据当前工作平面，编程格式如下所示：

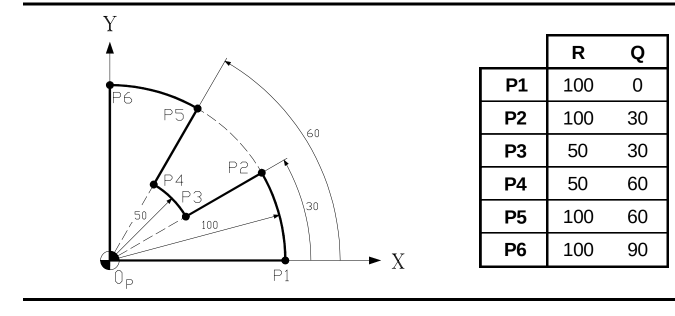
XY 平面 (G17)	G02/G03	R...	Q...	I...	J...
ZX 平面 (G18)	G02/G03	R...	Q...	I...	K...
YZ 平面 (G19)	G02/G03	R...	Q...	J...	K...

极坐标圆弧插补编程。





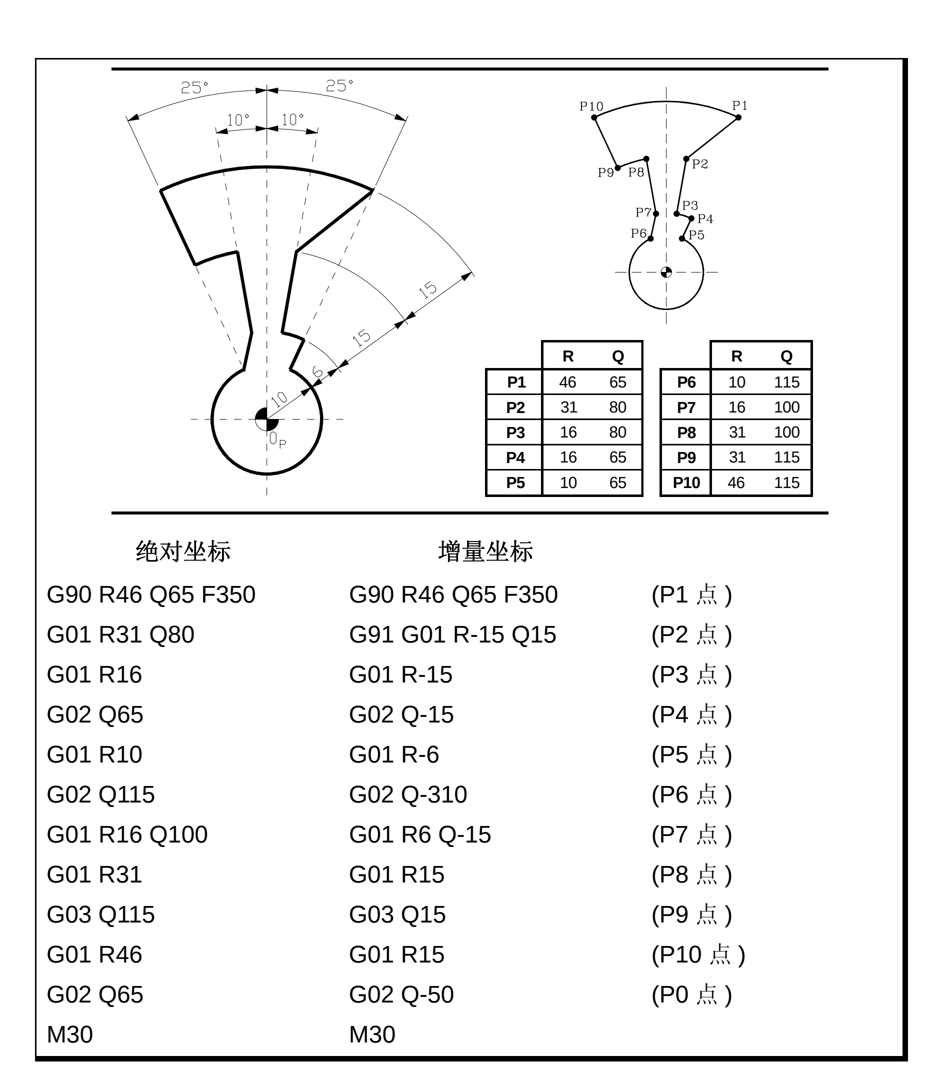
编程实例



	R	Q
P1	100	0
P2	100	30
P3	50	30
P4	50	60
P5	100	60
P6	100	90

绝对坐标	增量坐标	
G00 G90 X0 Y0 F350	G00 G90 X0 Y0 F350	(P0 点)
G01 R100 Q0	G91 G01 R100 Q0	(P1 点)
G03 Q30	G03 Q30	(P2 点)
G01 R50 Q30	G01 R-50	(P3 点)
G03 Q60	G03 Q30	(P4 点)
G01 R100 Q60	G01 R50	(P5 点)
G03 Q90	G03 Q30	(P6 点)
G01 R0 Q90	G01 R-100	(P0 点)
M30	M30	





	R	Q
P1	46	65
P2	31	80
P3	16	80
P4	16	65
P5	10	65

	R	Q
P6	10	115
P7	16	100
P8	31	100
P9	31	115
P10	46	115

绝对坐标	增量坐标	
G90 R46 Q65 F350	G90 R46 Q65 F350	(P1 点)
G01 R31 Q80	G91 G01 R-15 Q15	(P2 点)
G01 R16	G01 R-15	(P3 点)
G02 Q65	G02 Q-15	(P4 点)
G01 R10	G01 R-6	(P5 点)
G02 Q115	G02 Q-310	(P6 点)
G01 R16 Q100	G01 R6 Q-15	(P7 点)
G01 R31	G01 R15	(P8 点)
G03 Q115	G03 Q15	(P9 点)
G01 R46	G01 R15	(P10 点)
G02 Q65	G02 Q-50	(P0 点)
M30	M30	

6.3.4 临时极坐标原点移位至圆弧中心 (G31)

当用极坐标定义圆弧时，极坐标原点可以临时移位到圆弧中心处。

G31 临时极坐标原点移位至圆弧中心

功能 G31 临时移动极坐标原点到编程圆弧中心处。此功能仅在包含它的程序段内有效；一旦执行完该程序段，恢复极坐标原点位置。

此功能被加在圆弧插补 G2/G3 编程中。这种情况时，必须编写至少圆弧中心坐标中的一个坐标。



6.3.5 绝对坐标圆心 (G06/G261/G262)

当定义一个圆弧时，可以选择圆心位置是相对于圆弧起点或用绝对坐标定义。

编程

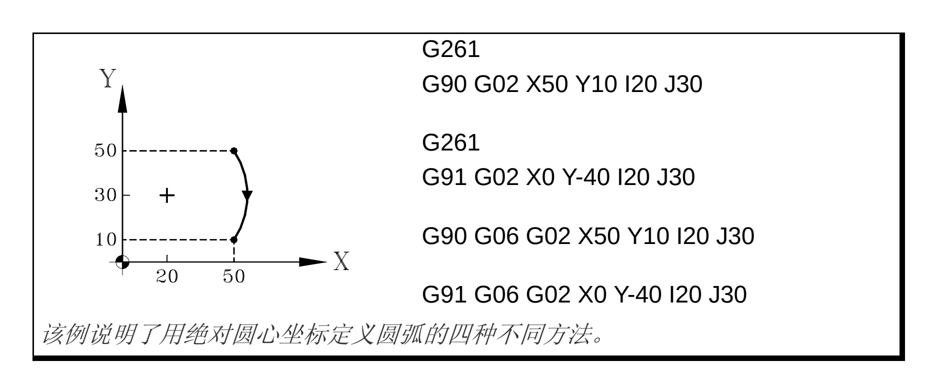
使用以下功能实现该选择：

G06	圆心用绝对坐标 (非模态)。
G261	圆心用绝对坐标 (模态)。
G262	圆心相对于起点。

G06-G261　绝对坐标圆心

当上述功能中的一个起作用时，数控系统认为圆弧中心坐标相对于当前坐标参考系原点 (工件零点、极坐标原点等)。

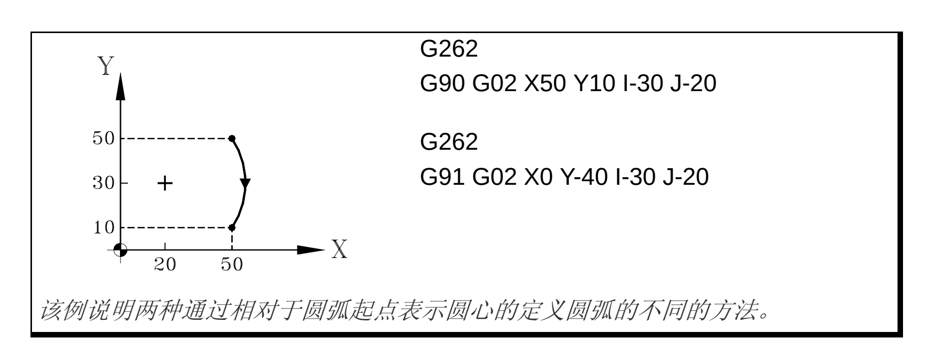
功能 G261 在程序中一直保持有效，但是 G06 仅在编写的程序段有效，因此它仅可以加在定义圆弧插补的程序段。



该例说明了用绝对圆心坐标定义圆弧的四种不同方法。

G262　相对于起点的圆心坐标

当该功能有效，数控系统认为圆心坐标是相对于圆弧起点的。



该例说明两种通过相对于圆弧起点表示圆心的定义圆弧的不同的方法。

功能特性

功能 G261 和 G262 是模态的，互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G262。



6.3.6 圆弧中心修正 (G264/G265)

为了执行编写的圆弧，数控系统计算初始半径和终点，这些必须一致。当不一致时，用圆弧中心修正，可以通过纠正圆弧中心执行编写的圆弧。

半径和定位校正圆弧中心之间的差异的允许误差由参数 CIRINERR 和 CIRINFAC 设置。

编程

圆弧中心修正可以用以下功能打开或关掉：

G264　　取消圆弧中心修正。

G265　　激活圆弧中心修正。

G264　**取消圆弧中心修正**

当初始半径和最终半径之间的差值在允许的误差范围内时，用初始点计算的半径完成圆弧。圆弧中心位置保留相同值。

如果初始半径和最终半径之间的差值超过允许的误差范围，发出相关错误报告。

G265　**激活圆弧中心修正**

如果初始半径和最终半径不同，数控系统试图在设置的误差范围内计算一个新的圆弧中心，使得执行的圆弧尽可能近似定义的圆弧。

计算误差范围在或不在公差之内，数控系统考虑两个值：

- 绝对误差 (半径差值)。
- 相对误差 (超过半径值的 %)。

如果其中任何一个值在 OEM 设置的公差之内，数控系统纠正圆弧中心位置。

如果数控系统不能使圆弧中心在这些限制范围内，数控系统将发出相关错误报告。

功能特性

功能 G264 和 G265 是模态的，彼此互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G265。



6.4 与前一路径相切圆弧 (G08)

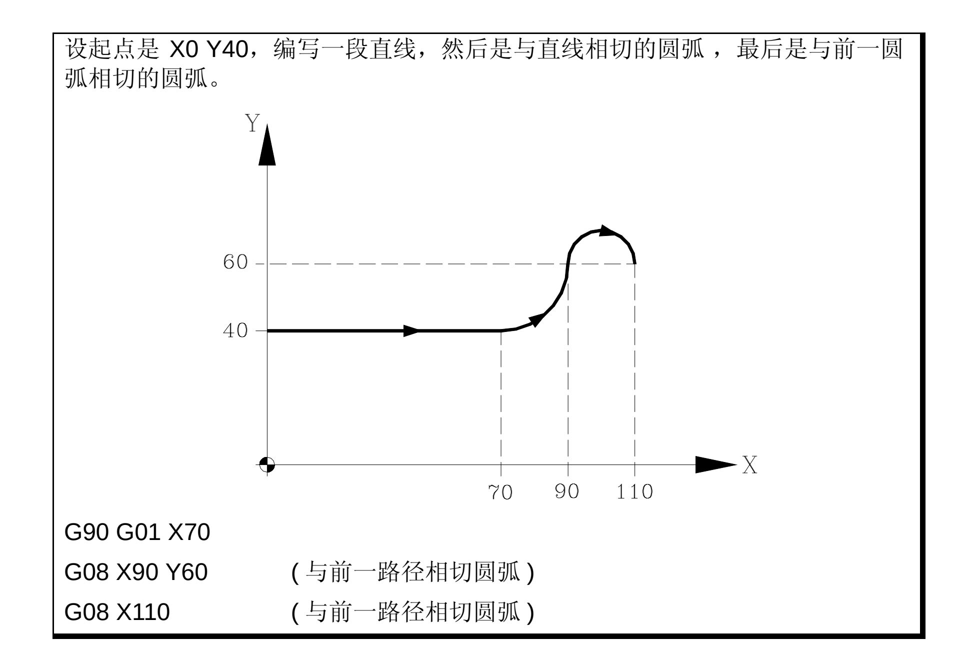
功能 G08 可以用于编写相切于前一路径的圆弧，不必编写圆弧中心坐标 (I, J 或 K)。

编程

仅需编写圆弧终点坐标，用沿工作平面轴的极坐标或笛卡尔坐标表示。

前一路径可以是直线或圆弧。

设起点是 X0 Y40，编写一段直线，然后是与直线相切的圆弧 ， 最后是与前一圆弧相切的圆弧。




```
G90 G01 X70
G08 X90 Y60        ( 与前一路径相切圆弧 )
G08 X110           ( 与前一路径相切圆弧 )
```

功能特性

功能 G08 是非模态的，因此，当编写一个与前一路径相切的圆弧时必须每次都编写 。完成此功能后，数控系统恢复之前运行的功能 G01, G02 或 G03 。

功能 G08 可以编写成 G8。

i

功能 G08 不能用于编写整圆，因为将会有无数种可能。



6.5 三点定义圆弧 (G09)

G09 可通过编写终点和中间点（圆弧起点是运动的开始点）定义圆弧 。换句话说，用中间点代替圆弧中心编程。

终点坐标

定义终点坐标可以用笛卡尔坐标或极坐标，可以用绝对坐标或增量坐标。

中间点坐标

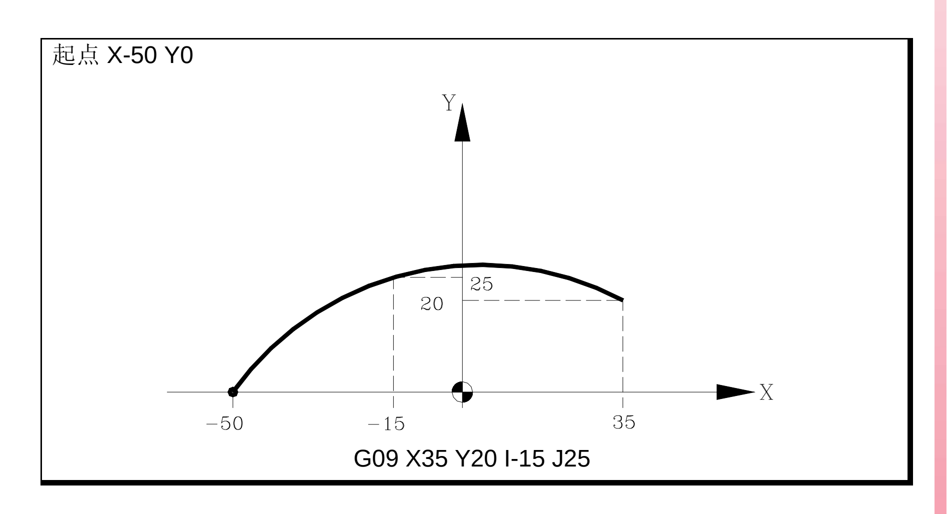
必须用笛卡尔坐标定义，根据当前平面用字母 "I", "J" 或 "K" 。

G17 G18 G19	字母 "I", "J" 和 "K" 分别对应 X, Y 和 Z 轴。
G20	字母 "I" 和 "J" 对应定义平面的横坐标轴和纵坐标轴。

这些坐标受功能 G90 和 G91 的影响。

编程格式根据当前的工作平面。在 XY 平面 是：

XY 平面 (G17)	G02/G03	X...	Y...	I...	J...
	G02/G03	R...	Q...	I...	J...

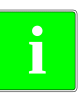


编写 G09 不需要编写运动方向 (G02 或 G03)。

功能特性

功能 G09 是非模态的，因此，当编写用三点定义的圆弧时必须每次编写该功能 。该程序段完成后，数控系统恢复之前运行着的功能 G01, G02 或 G03。

功能 G09 可以编写成 G9。



功能 G09 不可以用于编写整圆，因为三点必须是不同的。



6.6 螺旋插补 (G02/G03)

螺旋插补由工作平面的圆弧插补和编写的另一根轴的运动组成。

(A) (B)

(A) 简单螺旋插补

(B) 多转螺旋插补

螺旋插补编写在一段程序段中，它的圆弧插补用功能 G02, G03, G08 或 G09 编写。

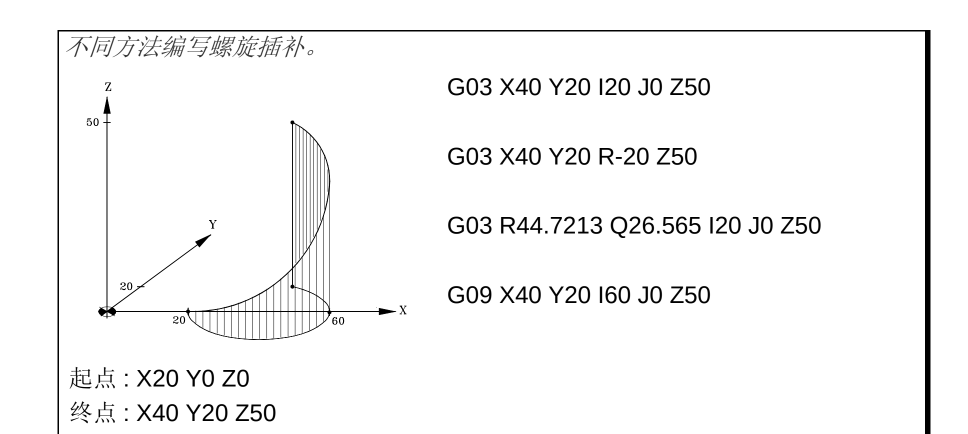
编程

简单螺旋插补

螺旋插补定义，编写当前平面的圆弧，然后是另一根轴的直线位移。

编程格式依据当前工作平面，在 XY 平面是 :

XY 平面 (G17)	G02/G03	X...	Y...	I...	J...	< 轴 >
	G02/G03	X...	Y...	R...	< 轴 >	
	G02/G03	R...	Q...	I...	J...	< 轴 >
	G08	X...	Y...	< 轴 >		
	G09	X...	Y...	I...	J...	< 轴 >



不同方法编写螺旋插补。

G03 X40 Y20 I20 J0 Z50

G03 X40 Y20 R-20 Z50

G03 R44.7213 Q26.565 I20 J0 Z50

G09 X40 Y20 I60 J0 Z50

起点 : X20 Y0 Z0
终点 : X40 Y20 Z50

编程

多转螺旋插补

如果螺旋插补产生多转，除了编写当前工作平面的圆弧插补和另一轴的运动外，必须编写螺距。



当定义圆弧插补的中心时，不需要定义在工作平面的终点坐标。数控系统将会根据螺旋的高度和螺距计算该点。

路径定义

螺距用和当前工作平面的三根轴对应的字母 "I", "J" 或 "K" 来 定义。

G17 G18 G19	螺距用字母 "K" (G17), "J" (G18) 或 "I" (G19) 定义。
G20	螺距用字母 "K" 定义。

编程格式依据当前工作平面，在 XY 平面是 :

XY 平面 (G17)	G02/G03	X...	Y...	I...	J...	< 轴 >	K...
	G02/G03	I...	J...	< 轴 >	K...		
	G02/G03	R...	Q...	I...	J...	< 轴 >	K...
	G08	X...	Y...	< 轴 >	K...		
	G09	X...	Y...	I...	J...	< 轴 >	K...

编写起点在 X0 Y0 Z0 的螺旋插补 .

```
G03 X0 Y0 I15 J0 Z50 K5

G03 R0 Q0 I15 J0 Z50 K5
```




6.7 等螺距螺纹加工 (G33)



对于螺纹加工，机床必须在主轴安装有旋转编码器.

当加工螺纹时，数控系统不能窜改主轴的运动。See **"6.8 刚性攻丝 (G63)"** on page 107.

尽管，这种类型的螺纹沿某根轴进行，数控系统允许同时对多根轴进行插补。除了可以加工单头螺纹，还可以加工多头螺纹，混合螺纹和螺纹修补。

螺纹可以沿任何轴进行，但是当不使用主轴时，使用的主轴必须和它同步。通过PLC 同步 (标志 SYNC)。

编程

螺纹编程用 G33 后跟螺纹终点坐标和螺距。切入角可随意定义，允许多头螺纹、混合螺纹和螺纹修补。

```
G33 X··Z I··J <Q1>
```

X··Z	终点坐标
I··K	螺距
Q1	可选择，切入角 。 如果没有编写，螺纹切入角采用 0º。

终点坐标

终点坐标可以用笛卡尔坐标或极坐标定义。可以用绝对坐标或增量坐标表示。

螺距

依据当前平面螺距用字母 "I", "J" 或 "K" 定义 。

G17 G18 G19	字母 "I", "J" 和 "K" 分别对应该通道的第一、第二和第三轴。
G20	字母 "I", "J" 和 "K" 对应定义平面的横坐标轴、纵坐标轴和垂直坐标轴 。

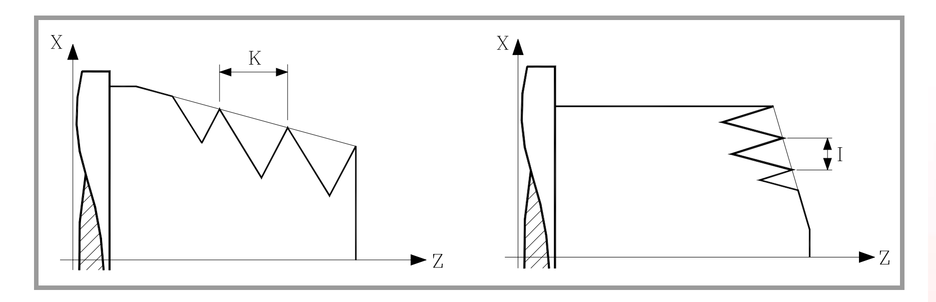
Z 轴在不同平面的螺纹加工实例。采用该通道的 X-Y-Z 轴系。

```
G17                  G18                  G19
G33 Z40 K2           G33 Z40 K2           G33 Z40 K2

G20 Z1 Y2 X3         G20 Y1 Z2 X3         G20 Y1 Z3 X2
G33 Z40 I2           G33 Z40 J2           G33 Z40 K2
```




当在螺纹加工中同时对多根轴进行插补时，不是在路径上定义螺距，而是在其中的一根轴上定义。



切入角

它表示螺纹起点的主轴角度位置 (±359.9999)。使用此参数可以加工多头螺纹。

编程时它是可选项，如果没有编写，将采用 0º (同编写 Q1=0)。

混合螺纹

当加工混合螺纹，只考虑第一根螺纹的切入角度。仅在激活 G33 后的第一根螺纹中考虑 Q1。参数 Q1 被忽略，一直到它取消又重新激活，并与角度同步。

执行注意事项

主轴零点搜索

如果主轴没有回零 (参考的), 当使用主轴时，将自动执行第一个 G33 。如果主轴是主轴，并且没有回过零，将会发出警告。

进给率和速度

螺纹进给率取决于编程主轴转速和螺距 (进给率 = 主轴速度 x 螺距)。

螺距在 100% 进给率 F 和主轴速度 S 执行，并且这些值不能从数控系统的操作面板或通过 PLC 进行修调。

功能特性

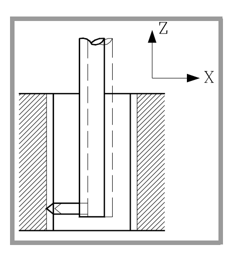
功能 G33 是模态的，和功能 G00, G01, G02, G03 以及 G63 互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G00 或 G01 由机床厂商通过参数 IMOVE 设置。



铣削编程实例

单头螺纹加工



单一路径加工如下电子螺纹。

位置 : X30 Y30 Z0
深度 : 30mm
螺距 : 1.5mm

```
S100 M03
G01 G90 X30 Y30 Z0
G33 Z-30 K1.5
M19 S0                  ( 主轴定位 )
G91 X3                  ( 刀具退离已加工表面 )
G90 Z10                 ( 退出孔 )
```

编写主轴转速 100 rpm 和螺距 1.5 mm，所以最后加工进给率将是 150 mm/min (螺距进给率)。

多头螺纹

从三个不同切入点加工类似前面的简单螺纹，第一个在开始 20º。

```
S100 M03
G01 G90 X30 Y30 Z0
G33 Z-30 K1.5 Q1=20     ( 第一头螺纹 )
M19 S0
G91 X3
G90 Z10
S100 M03
G33 Z-30 K1.5 Q1=140    ( 第二头螺纹 )
M19 S0
G91 X3
G90 Z10
S100 M03
G33 Z-30 K1.5 Q1=260    ( 第三头螺纹 )
M19 S0
G91 X3
G90 Z10
S100 M03
M30
```

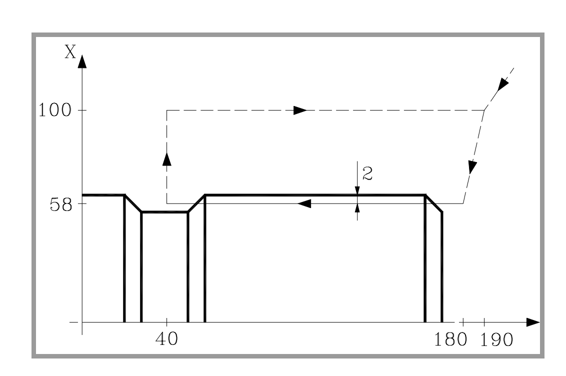



车削编程实例

X 轴以半径编程方式的编程实例。

纵向螺纹

单一路径加工圆柱形螺纹，深度 2 mm，螺距 5 mm。

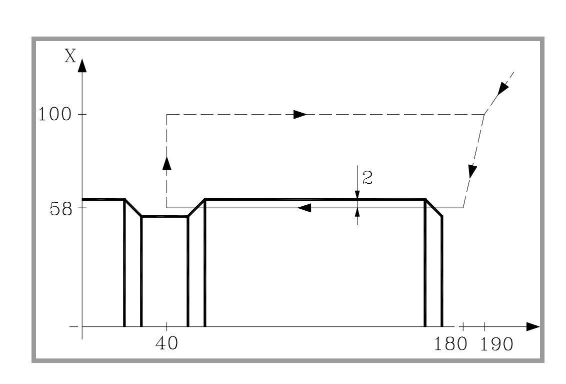



```
S100 M03
G00 G90 X200 Z190
X116 Z180
G33 Z40 K5
G00 X200
Z190
```

编写主轴转速100 rpm和 螺距5mm，最终加工进给率将是500 mm/min (螺距进给率).

多头纵向螺纹

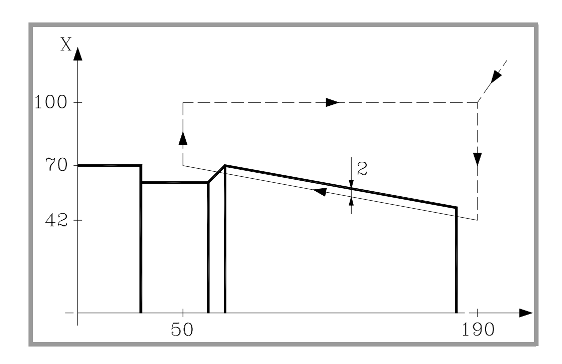
类似前面加工一个简单螺纹，另两个简单螺纹的切入角彼此转过 180º。




```
S100 M03
G00 G90 X200 Z190
X116 Z180
G33 Z40 K5 Q1=0
G00 X200
Z190
X116 Z180
G33 Z40 K5 Q1=180
G00 X200
Z190
```

锥螺纹

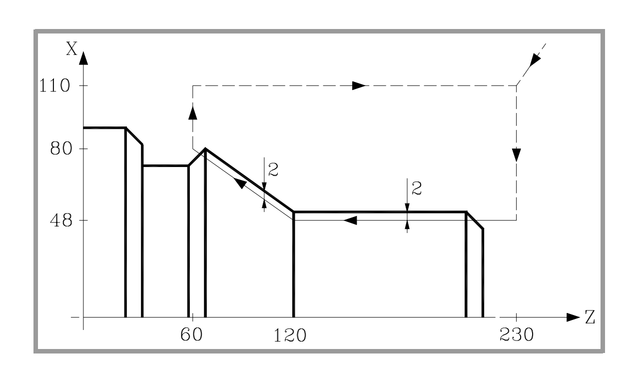
单一路径加工锥螺纹，深 2 mm，螺距 5 mm。




```
S100 M03
G00 G90 X200 Z190
X84
G33 Z140 Z50 K5
G00 X200
Z190
```

混合螺纹

加工一段纵向螺纹接一段锥螺纹的混合螺纹，深度 2 mm，螺距 5 mm。




```
S100 M03
G00 G90 G05 X220 Z230
X96
G33 Z120 Z50 K5
G33 X160 Z60 K5
G00 X220
Z230
```



6.8 刚性攻丝 (G63)

对于刚性攻丝，机床主轴上必须安装旋转编码器。

刚性攻丝时，数控插入纵向轴相关主轴的运动。

编程

定义一个刚性攻丝，编写功能 G63 后跟螺纹终点坐标，坐标定义可以用笛卡尔坐标或极坐标。数控系统依据进给速度 F 和主轴转速 S 计算螺距。

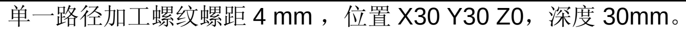
功能 G63 以编写的速度 S 指示的方向启动主轴，忽略当前功能 M3, M4, M5 或 M19。负旋转速度仅能在 G63 有效时编写。

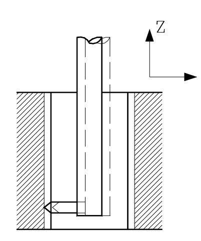
```
...
G94 F300
G01 G90 X30 Y30 Z50
G63 Z20 S200
...
```

螺距将是 : $\frac{F}{S} = \frac{300}{200} = 1{,}5\text{mm}$

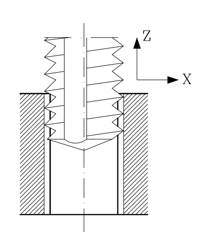
因为 G63 在攻丝后不能自动将刀具退出螺纹孔，为了退出刀具必须编写一个使主轴反方向旋转的反向螺纹加工 (通过改变 S 速度符号) 如果螺纹使用起锥器，也可以通过主轴定向 (M19) 和从螺纹分离起锥器的方法退出刀具。

单一路径加工螺纹螺距 4 mm ， 位置 X30 Y30 Z0，深度 30mm。






```
G94 F400
G01 G90 X30 Y30 Z0
G63 Z-30
M19 S0
G91 X3
G90 Z10
```




```
G94 F400
G01 G90 X30 Y30 Z0
G63 Z-30 S100
G63 Z0 S-100
G01 Z10
```




多头螺纹

这种类型的螺纹，可以加工具有多个切入点的螺纹。在每个螺纹加工操作前必须定义每个切入点的位置。

```
...
G90 G01 X0 Y0 Z0 F150
M19 S0                      ( 第一头螺纹在 0º 切入 )
G63 Z-50 S150                 ( 攻螺纹 )
G63 Z0 S-150                  ( 退回 )
M19 S120                    ( 第二头螺纹在 120º 切入 )
G63 Z-50 S150
G63 Z0 S-150
M19 S240                    ( 第三头螺纹在 240º 切入 )
G63 Z-50 S150
G63 Z0 S-150
...
```

三头螺纹，深度 50mm ， 螺距 1mm 。

执行注意事项

主轴速度运行方式

依据定义的转速，操作将如下进行：

- 如果在 G63 有效时定义了螺纹加工速度，速度将保持有效，一直到 G63 被撤销，然后会恢复螺纹加工开始前的有效速度。
- 如果没有特别定义螺纹加工速度，将执行当时有效的速度。

主轴旋转方向由编写的 "S" 速度的符号决定，忽略有效功能 M3, M4, M5 或 M19 。编写这些功能中的一个将会撤销 G63。

进给率运行方式

当加工刚性螺纹，进给率可以在 0% 到 200% 之间变化，用数控系统操作面板的进给倍率按键或通过 PLC 实现。数控系统将改变主轴速度，以保持进给轴和主轴的插补。

刚性攻丝和刀具检查模式

当中断刚性攻丝加工，访问刀具检查模式时，可以手动操作包括在螺纹加工中的轴（仅在手动模式下）。当移动轴时，插补轴也一起运动；主轴用来加工螺纹。如果刚性攻丝包括多根轴，当移动其中的一根轴时，所有其它包括在螺纹加工中的轴也将一起运动。

可以让轴从螺纹孔移进或移出，直到按下相应的软键。轴以编写的 F 速度运动，除了当轴或主轴超过允许的最大进给率时 (参数 MAXMANFEED)，在这种情况时，进给率将被限制至最大进给率。

在刀具检查时主轴手动键不起作用。仅可能将刚性攻丝加工中包括的轴中的一根通过手动方式退出螺纹孔。对主轴不能编写功能 M3, M4, M5 和 M9 ；它们被忽略。



重新定位时，当在软键菜单中选择加工螺纹的一根轴时，将会移动所有的轴和包含在螺纹加工中的主轴。

功能特性

功能 G63 是模态的，和功能 G00, G01, G02, G03 以及 G33 互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G00 或 G01 由机床厂商通过参数 IMOVE 设置。



6.9 手动干涉 (G200/G201/G202)

使用这些功能，可以通过编程激活手动模式；换句话说，甚至在执行程序时也可以手动执行轴。运动可以用手轮或手动键 （增量或连续手动）执行。

编程

相关手动干涉功能如下：

G200	专用手动干涉。
G201	附加手动干涉生效。
G202	附加手动干涉撤销。

专用手动干涉和附加手动干涉之间的区别是，专用手动干涉 (G200) 中断程序执行去激活手动模式，但是附加手动干涉 (G201) 在执行程序时可以实现轴的手动。

进给率运行方式

在手动干涉期间，手动运动的进给率不受当前 F 约束，可以由操作者用高级语言指令定义 (如本手册中的 “16 语句与指令” 章所述)；不同的进给率可以设置给每个工作模式 (增量或连续手动)。如果没有定义，运动执行机床参数设置的进给率 。

进给率可以在 0% 到 200% 之间变化，用数控系统操作面板上的进给率倍率开关，可以同时影响编写的 F 和手动干涉速度。

编程特性

功能 G201, G202 (模态) 和 G200 (非模态) 互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G202。



6.9.1 附加手动操作 (G201/G202)

有此功能，可以在执行程序时用手轮或手动键 （连续或增量）手动操作轴。

可以应用于机床的任何轴。不能应用于主轴，即使能工作在点动模式。

G201 附加手动干涉生效

激活附加手动干涉，编写 G201 后跟受它影响的轴，使用指令 "#AXIS[< 轴 >]"。

功能 G201 必须总是后跟指令 "#AXIS"，至少定义一根轴 。

G202 附加手动干涉撤销

撤销附加手动干涉，编写 G202 后跟待撤销的轴，使用指令 "#AXIS[< 轴 >]"。

单独编写 G202 ，对所有轴撤销手动干涉。

```
...
N100 G71 G90 X0 Y0 F400
N110 G201 #AXIS [X, Z]        ( 在 X-Z 轴激活附加手动干涉 )
N120 G01 X100 Y50             ( 可以手动 X-Z 轴 )
N130 G202 #AXIS [X]           ( 在 X 轴撤销手动干涉 )
N140 G01 X50 Y150             (Z 轴可以手动 )
N150 G202 #AXIS [Z]           ( 在 Z 轴撤销手动干涉 )
...
N200 G201 #AXIS [X, Y, Z]     ( 在 X-Y-Z 轴激活附加手动干涉 )
N220 G01 X100 Y50             ( 可以手动 X-Y-Z 轴 )
N230 G202                     ( 对所有轴撤销手动干涉 )
...
```

注意事项

轴机床参数 MANFEEDP, IPOFEEDP, MANACCP, IPOACCP 决定各类运动 （手动或自动）的许可进给率和最大加速度。如果两个附加手动干涉进给率超过 100%，用户必须保证两个运动不是在同一根轴同时进行，因为可能引起动力过载。



6.9.2 专用手动干涉 (G200)

有此功能，轴可以用手轮或手动键 （连续或增量）手动方式操作，中断程序执行。

撤销手动干涉并重新继续执行程序，按 [CYCLE START][a] 键。

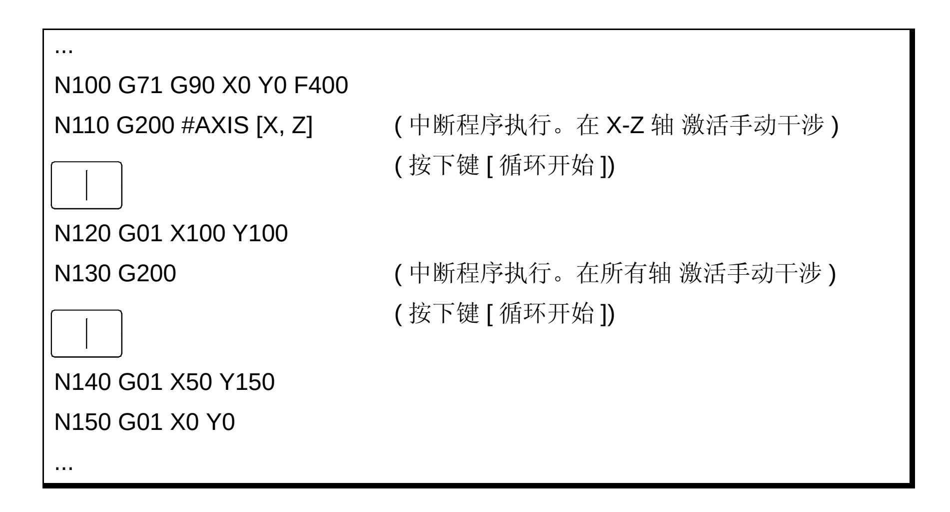
(a)

可以应用到机床的任意轴。不能应用到主轴，即使能工作在点动模式。

G200 专用手动干涉

激活专用手动干涉，编写 G200 后跟影响的轴，用指令 "#AXIS[<axes>]"。

单独编写 G200 应用手动干涉到所有轴 。



```
...
N100 G71 G90 X0 Y0 F400
N110 G200 #AXIS [X, Z]        ( 中断程序执行。在 X-Z 轴 激活手动干涉 )
                              ( 按下键 [ 循环开始 ])
N120 G01 X100 Y100
N130 G200                     ( 中断程序执行。在所有轴 激活手动干涉 )
                              ( 按下键 [ 循环开始 ])
N140 G01 X50 Y150
N150 G01 X0 Y0
...
```

注意事项

如果在圆弧插补前执行手动干涉，且包括在圆弧插补中的轴中的一根是手动，将会发送错误报告，指示圆弧插补编程错误或可以执行编程之外的圆弧。



几何知识

7

7.1 方角 (G07/G60)

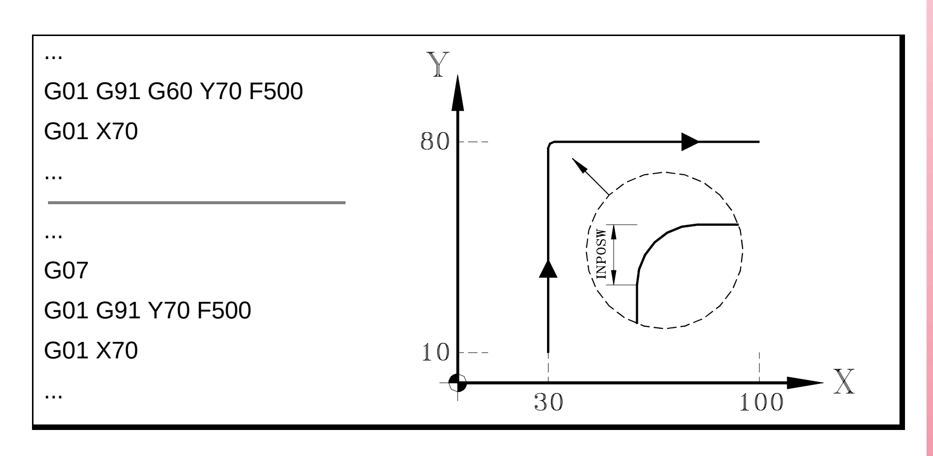
当工作在方角模式时，数控系统直到轴到达编程位置才开始执行下一个运动。当轴定位在数控参数INPOSW设置的“到位”区域内，数控系统认为已经到达编程位置。

编程

方角加工模式可以用以下不同的两个功能激活：

G07	方角 (模态)。
G60	方角 (非模态)。

功能 G07 在程序始终保持有效。但是 G60 仅影响它所在的程序段；因此，仅能添加到包含运动的程序段。



理论和实际轮廓重合，得到方角，如图所示。

功能特性

功能 G07 是模态的，和功能 G05, G50, G60, G61 以及 HSC 模式不兼容。

功能 G60 是非模态的。执行后，数控系统恢复先前有效的功能 G05, G07, G50 或 HSC 模式 。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G05, G07 或是 G50 取决于对机床参数 ICORNER 的设置。

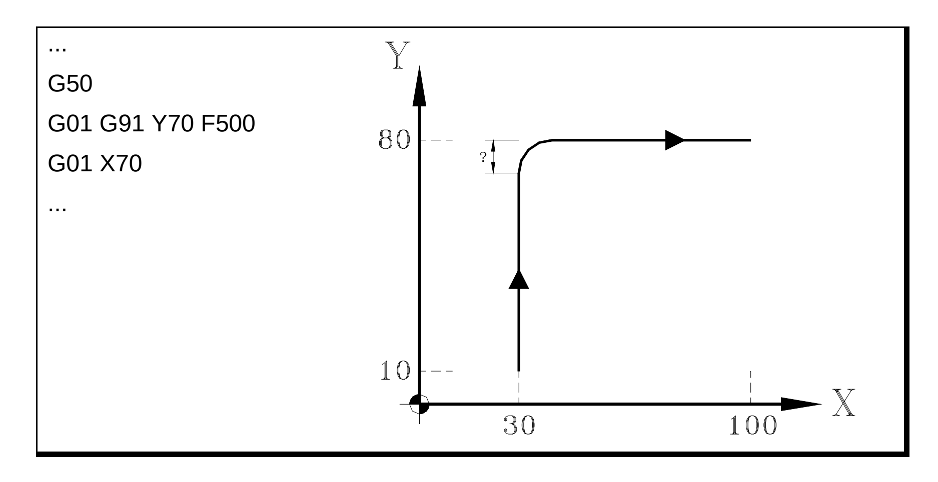


7.2 半圆角 (G50)

当工作在半圆角模式下时，一旦当前运动的理论插补完成，不需要等轴到位，数控系统开始执行下一个运动。从编程位置到下一个运动开始的位置之间的距离取决于轴的进给率。

编程

半圆角加工模式可以用功能 G50 编程激活。



此功能可加工如图所示的圆角。

功能特性

功能 G50 是模态的，和 G05, G07, G60, G61 以及 HSC 模式不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G05, G07 或是 G50 取决于对机床参数 ICORNER 的设置。



7.3 可控圆角，半径过渡 (G05/G61)

当工作在圆角模式下时，可以控制编程轮廓的拐角。机床如何来执行则取决于选择的圆角类型。

编程

圆角类型通过指令 "#ROUNDPAR" 选择，它将保持有效，直到选择不同的圆角类型。本章的 *"7.3.1 圆角类型"* 部分对可用的各类圆角作了详细说明。

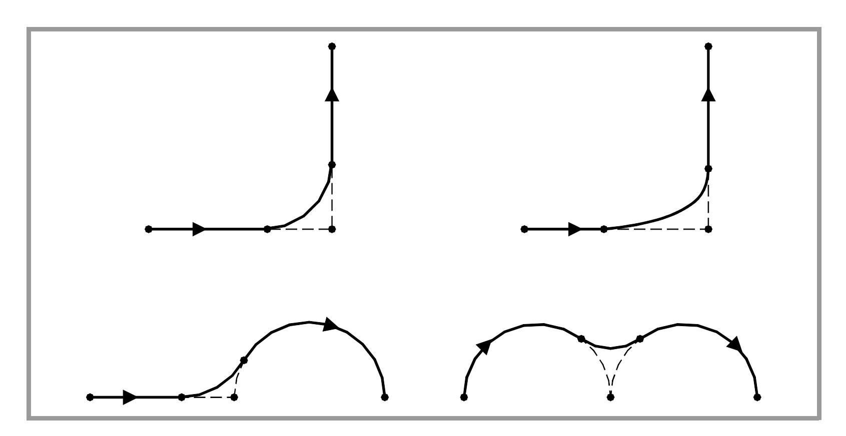
在选择圆角类型之后，可以用以下功能通过编程来激活。

G05	可控圆角，半径过渡 (模态)。
G61	可控圆角，半径过渡 (非模态)。

功能 G05 在程序中始终有效，但是功能 G61 仅影响它所在的程序段；因此，仅能添加在包含运动的程序段。

注意事项

此操作应用在任何拐角，不管是定义在直线之间、直线和圆弧还是圆弧之间的路径。



沿曲线路径加工拐角，不是圆弧。曲线的形状取决于选择的圆角类型和有关轴的动力学条件 （进给率和加速度）。

功能特性

功能 G05 是模态的， 和 G07, G50, G60, G61 以及 HSC 模式不兼容。

功能 G61 是非模态的。执行后，数控系统恢复先前作用的功能 G05, G07, G50 或 HSC 模式。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G05, G07 或是 G50 取决于对机床参数 ICORNER 的设置。

7.3.1 圆角类型

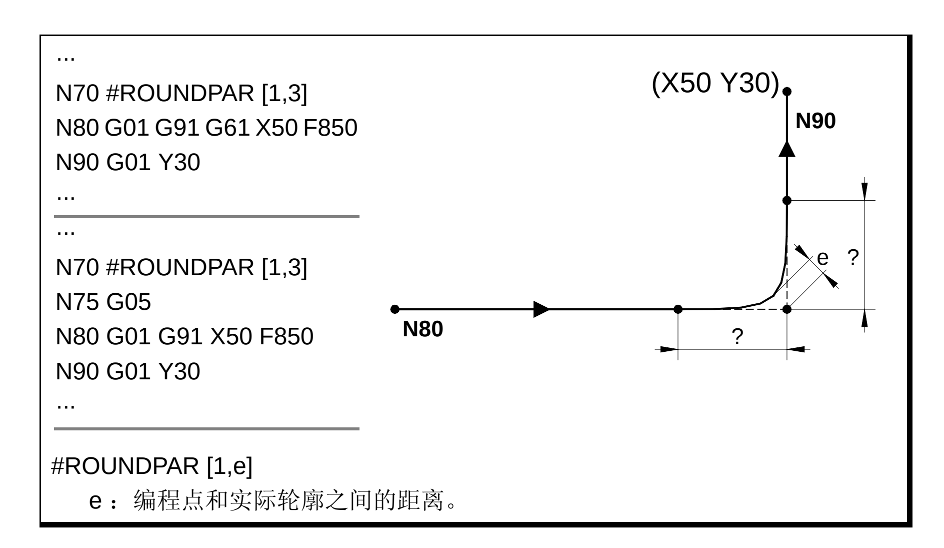
有 5 种不同的圆角轮廓类型。前 4 种执行不同的圆形拐角，但是最后一种执行方角。最后一种针对特殊机床 (激光，水射流，等)，用它可避免 “燃烧” 拐角，因而不推荐用于铣床。

圆形拐角选择和定义通过和指令 "#ROUNDPAR" 关联的参数进行。该指令可以关联最多 6 个参数，这些参数的含义取决于选择的圆角类型。

类型 1 #ROUNDPAR [1,e]

设置从圆角编程点和结果轮廓之间允许的最大偏差。

通过提供优先权给机床动力条件 （进给率和加速度）来圆角。执行机床操作尽量接近编程点但不超过编程偏差，不需要减少编程进给率 "F"。



从编程点到的圆角开始和结束点之间的距离会自动计算，不能大于程序段编写的路径的一半。两个距离是相等的，除了当其中的一个被限制到编程路径的一半时。

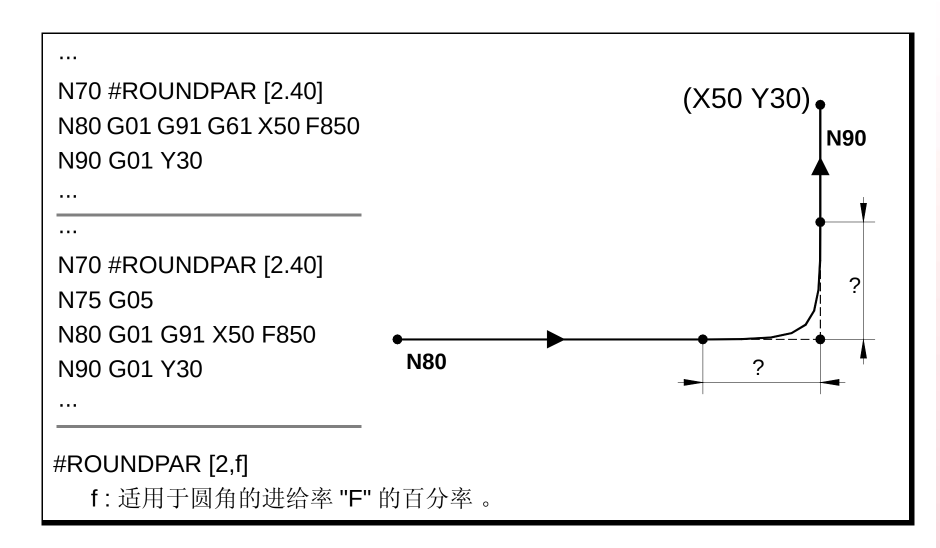
对于这种类型的圆角，仅使用指令 "#ROUNDPAR" 的前两个参数，所以，不需包括所有参数。



类型 2 #ROUNDPAR [2,f]

设置有效进给率 "F" 的百分率，用于执行圆角。

执行圆角尽量接近编程点，可以按照设置的进给率百分率加工圆角。

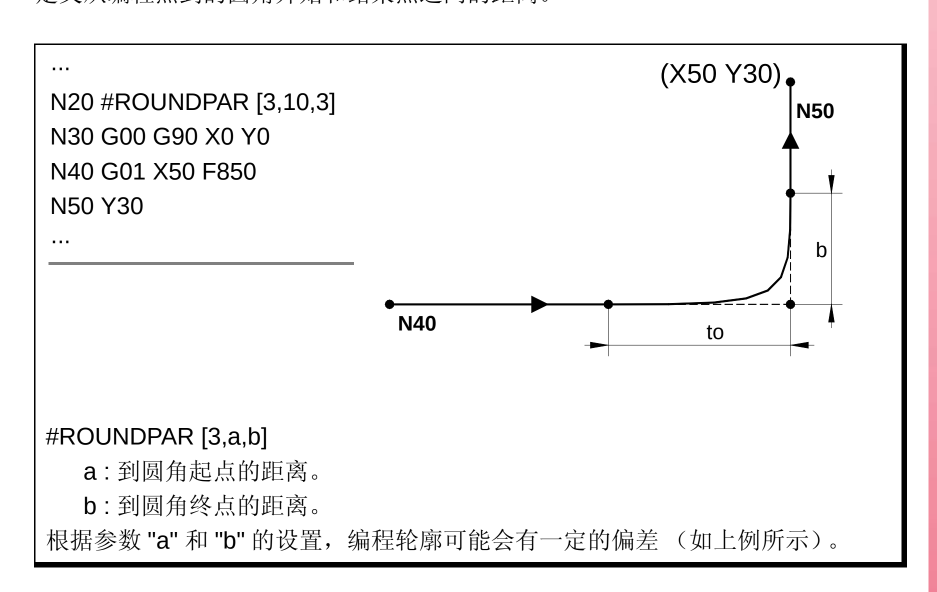


从编程点到的圆角开始和结束点之间的距离会自动计算，不能大于程序段编写的路径的一半。两个距离是相等的，除了当其中的一个被限制到编程路径的一半时。

对于这种类型的圆角，仅使用指令 "#ROUNDPAR" 的前两个参数，所以，不需包括所有参数。

类型 3 #ROUNDPAR [3,a,b]

定义从编程点到的圆角开始和结束点之间的距离。



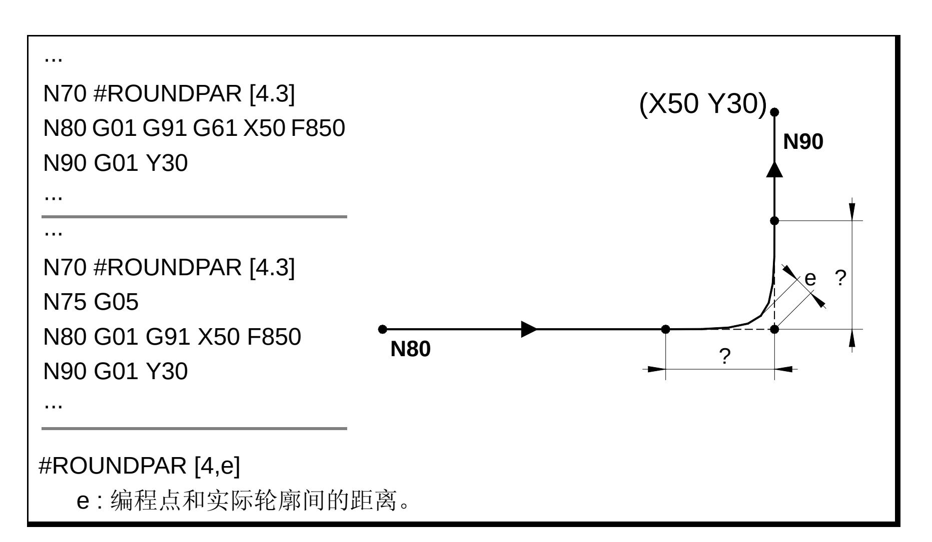
对于这种类型的圆角，仅使用指令 "#ROUNDPAR" 的前三个参数，所以，不需包括所有参数。



类型 4 #ROUNDPAR [4,e]

设置编程点和圆角结果轮廓之间允许的最大偏差。

通过提供优先权给机床动力条件 （进给率和加速度）来圆角。需要时减小进给率 "F"，执行编写的加工操作。

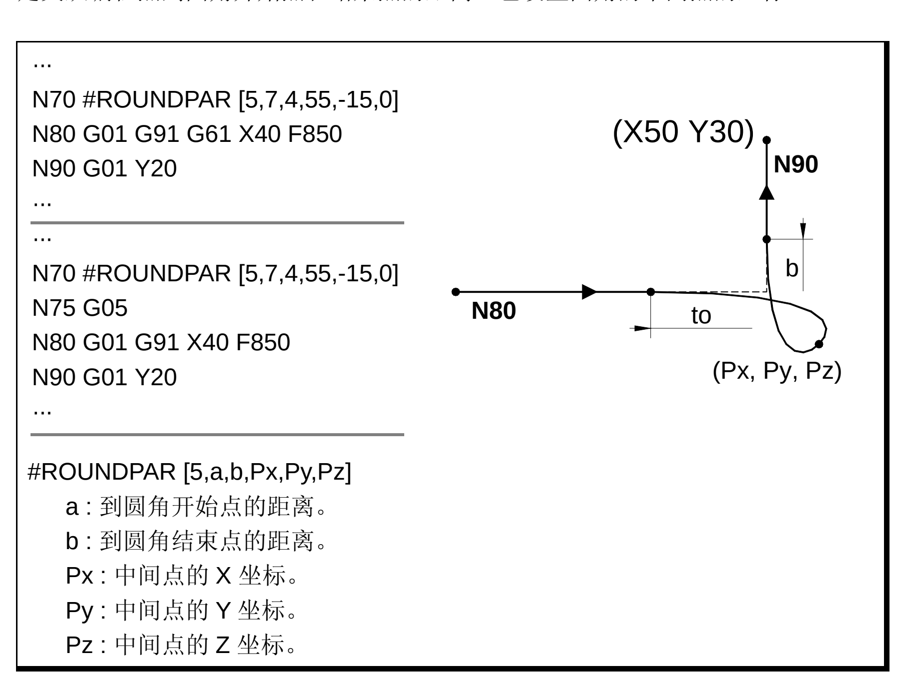


从编程点到的圆角开始和结束点之间的距离会自动计算，不能大于程序段编写的路径的一半。两个距离是相等的，除了当其中的一个被限制到编程路径的一半时。

对于这种类型的圆角，仅使用指令 "#ROUNDPAR" 的前两个参数，所以，不需包括所有参数。

类型 5 #ROUNDPAR [5,a,b,Px,Py,Pz]

定义从编程点到圆角开始点和结束点的距离。也设置圆角的中间点的坐标。



对于这种类型的圆角，仅使用指令 "#ROUNDPAR" 的前六个参数。

这种类型的圆角，曲线的形状取决于中间点的位置和从编程点到圆角起点和终点的距离。



```
...
G92 X0 Y0
G71 G90
#ROUNDPAR [5,-30,-30,55,-5,0]
G01 G61 X50 F850
N90 G01 Y40
...
```

"a" 和 "b" 距离是负值，在每根轴上的绝对值大于编程点到中间点的距离 （大约 4 倍）。

```
...
G92 X0 Y0
G71 G90
#ROUNDPAR [5,-5,-5.65,-15.0]
G01 G61 X50 F850
G01 Y40
...
```

"a" 和 "b" 距离是负值，在每根轴上的绝对值小于编程点到中间点的距离。

```
...
G92 X0 Y0
G71 G90
#ROUNDPAR [5,5,5,65,-15,0]
G01 G61 X50 F850
G01 Y40
...
```

"a" 和 "b" 距离是正值 。



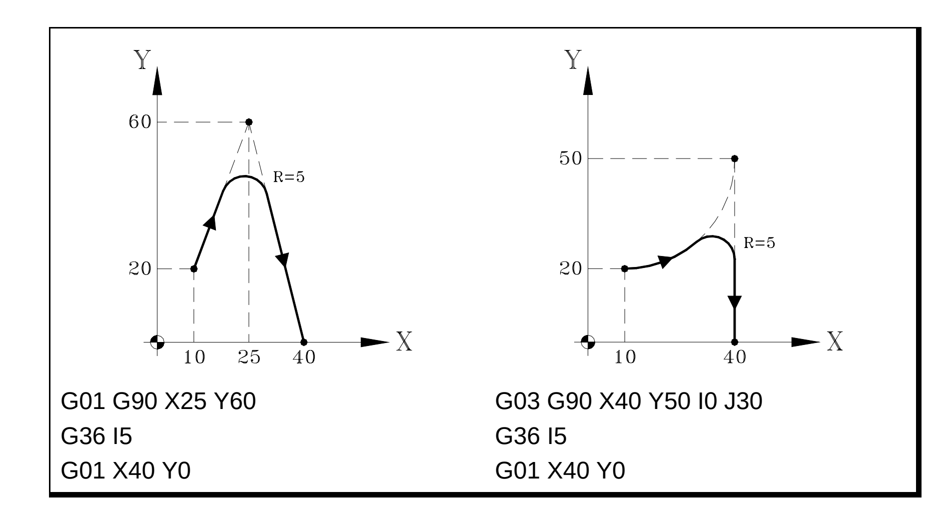
7.4 圆角，半径过渡 (G36)

G36 可以使用特定半径进行圆角，而不用去计算中心或圆弧起点和圆弧终点。

编程

过渡圆弧定义必须编写在两个路径之间。路径可以是直线和直线、直线和圆弧、圆弧和直线。

编程格式 "G36 I< 半径 >"，半径值编程用单位毫米或英寸，这取决于当前的单位值。



注意事项

圆弧半径 "I" 值保持有效，一直到编写另一个值 ，因此，不需要在连续的圆弧过渡操作中编写同样的半径值。

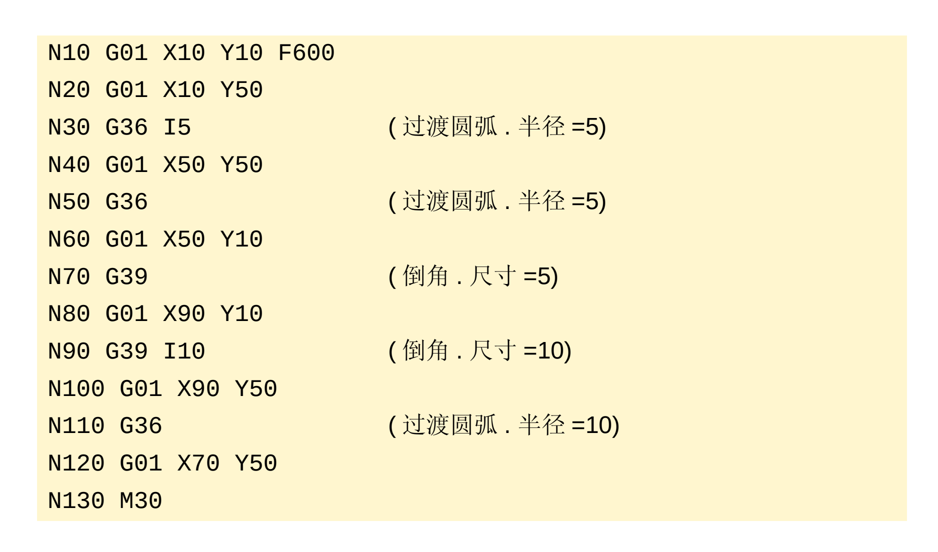
过渡圆弧半径 "I" 值用在以下功能 :

G37 (切入) 作切入半径。

G38 (切出) 作切出半径。

G39 (倒角) 作倒角尺寸。

这意味着当编写其中一条指令时，在 G36 中设置的圆弧半径将成为切入半径、切出半径或倒角尺寸，反之亦然。



```
N10 G01 X10 Y10 F600
N20 G01 X10 Y50
N30 G36 I5               ( 过渡圆弧 . 半径 =5)
N40 G01 X50 Y50
N50 G36                  ( 过渡圆弧 . 半径 =5)
N60 G01 X50 Y10
N70 G39                  ( 倒角 . 尺寸 =5)
N80 G01 X90 Y10
N90 G39 I10              ( 倒角 . 尺寸 =10)
N100 G01 X90 Y50
N110 G36                 ( 过渡圆弧 . 半径 =10)
N120 G01 X70 Y50
N130 M30
```




编写的过渡圆弧进给率取决于随后编写的运动的类型：

- 如果下一个运动是 G00，过渡圆弧将执行 G00。
- 如果下一个运动是 G01, G02 或 G03，过渡圆弧将执行编写在过渡圆弧定义段的进给率。如果没有编写进给速度，以当前的进给率执行。

```
N10 G01 G94 X10 Y10 F600
N20 G01 X10 Y50
N30 G36 I5                  ( 以 G00 倒角 )
N40 G00 X50 Y50
N50 G36                     ( 倒角， F=600mm/min。 )
N60 G01 X50 Y10
N70 G36 F300                ( 倒角，F=300mm/min。 )
N80 G01 X90 Y10 F600
N90 M30
```

在两个路径之间定义平面转换，定义的圆弧过渡在第二路径定义的平面被执行。

```
N10 G01 G17 X10 Y10 Z0 F600
N20 X10 Y50                 (X-Y 平面 )
N30 G36 I10
N40 G18                     (Z-X 平面，圆弧过渡在此平面执行。 )
N50 X10 Z30
N60 M30
```

功能特性

功能 G36 是非模态的，所以，每次圆弧过渡时必须编写。

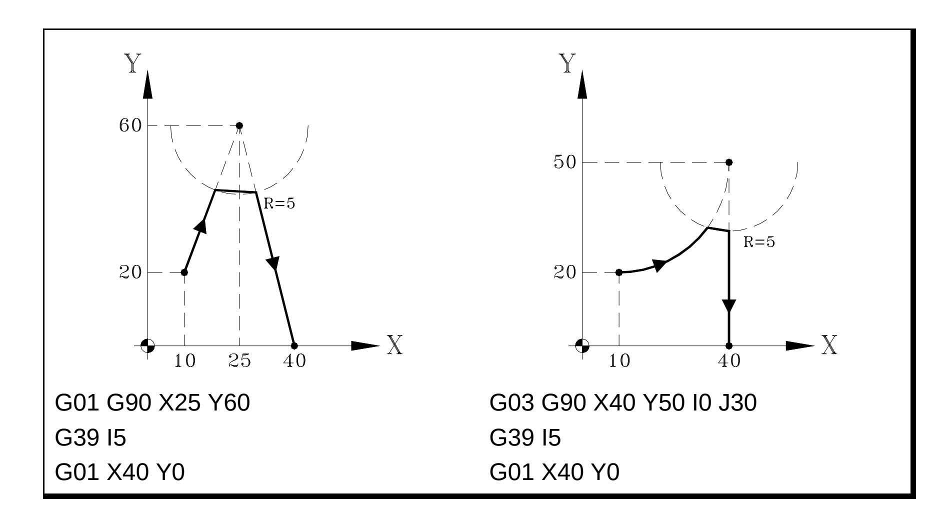
7.5 倒角 (G39)

功能 G39 可以用于插入一个特定尺寸的倒角，不需要计算交叉点。

编程

倒角必须定义在定义待拐角点的两路径之间。两路径可以是直线接直线、直线接圆弧、圆弧接圆弧。

编程格式 "G39 I< 尺寸 >"，根据当前的单位制，尺寸单位用毫米或英寸。



注意事项

倒角尺寸 "I" 值保持有效，直到编写另一个值，所以，不需要在连续的同尺寸的倒角操作中编写。

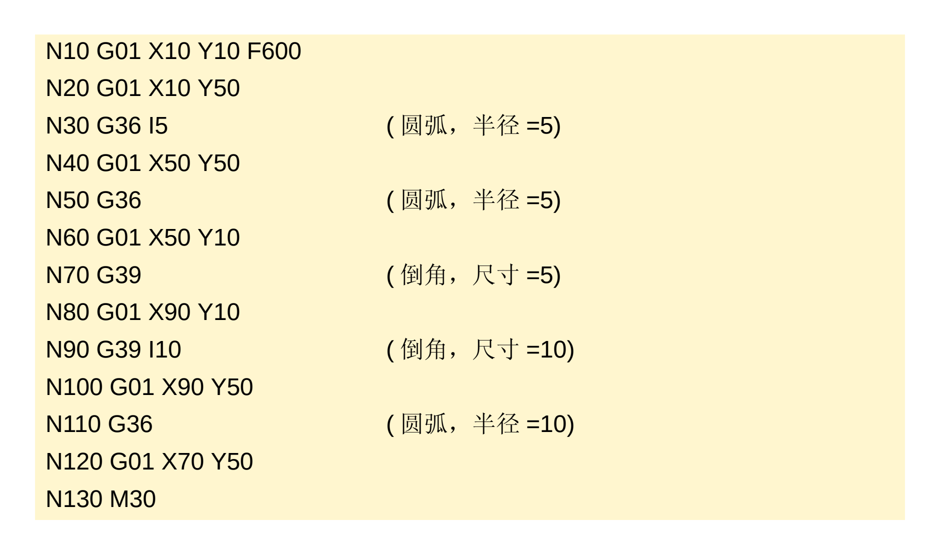
倒角尺寸 "I" 值也用于功能：

G36 (圆角) 作圆弧半径 。

G37 (切线切入) 作切入半径。

G38 (切出) 作切出半径。

这意味着当编写其中一条指令时，在 G39 中设置的倒角尺寸将成为切入半径、切出半径或圆弧半径，反之亦然。



```
N10 G01 X10 Y10 F600
N20 G01 X10 Y50
N30 G36 I5                    ( 圆弧，半径 =5)
N40 G01 X50 Y50
N50 G36                       ( 圆弧，半径 =5)
N60 G01 X50 Y10
N70 G39                       ( 倒角，尺寸 =5)
N80 G01 X90 Y10
N90 G39 I10                   ( 倒角，尺寸 =10)
N100 G01 X90 Y50
N110 G36                      ( 圆弧，半径 =10)
N120 G01 X70 Y50
N130 M30
```




编写的倒角进给率取决于随后编写的运动类型：

- 如果下一个运动是 G00，过渡圆弧将执行 G00。
- 如果下一个运动是 G01, G02 或 G03，过渡圆弧将执行编写在过渡圆弧定义段的进给率。如果没有编写进给速度，以当前的进给率执行。

```
N10 G01 G94 X10 Y10 F600
N20 G01 X10 Y50
N30 G39 I5                  ( 以 G00 倒角 )
N40 G00 X50 Y50
N50 G39                     ( 倒角， F=600mm/min.)
N60 G01 X50 Y10
N70 G39 F300                ( 倒角， F=300mm/min.)
N80 G01 X90 Y10 F600
N90 M30
```

在两个路径之间定义平面转换，定义的圆弧过渡在第二路径定义的平面被执行。

```
N10 G01 G17 X10 Y10 Z0 F600
N20 X10 Y50                 (X-Y 平面 )
N30 G39 I10
N40 G18                     (Z-X 平面，倒角在此平面执行 )
N50 X10 Z30
N60 M30
```

功能特性

功能 G39 是非模态的，所以，每次倒角过渡时必须编写。



7.6 切入 (G37)

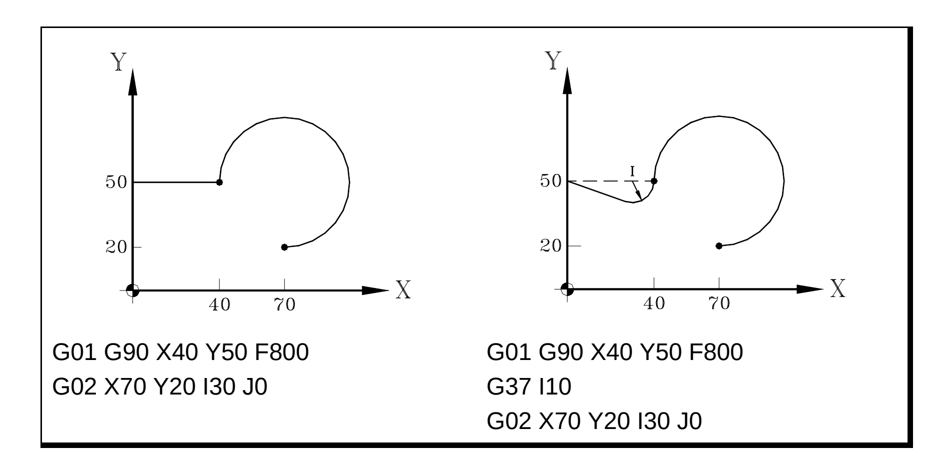
功能 G37 可以让刀具开始切入加工，而不必计算交点坐标。

编程

切线切入必须单段编写，该段后的路径被修改；路径必须是直线 (G00 或 G01)。

编程格式 "G37 I< 半径 >"，半径值单位为毫米或英尺，取决于当前的单位制。

在切入前的直线路径长度必须等于或大于两倍的切入半径。同样地，半径必须是正的，并且当带刀具半径补偿工作时，其必须大于刀具半径。



注意事项

切入半径 "I" 值保持有效，直到编写另一个值，所以，不需要在连续的同尺寸的切入操作中编写。

倒角尺寸 "I" 值也用于功能：

G36 (圆角) 作圆弧半径 。

G38 (切出) 作切出半径。

G39 (倒角) 作倒角尺寸。

这意味着当编写其中一条指令时，在 G37 设置的切入半径将成为切出半径，圆弧半径或倒角尺寸，反之亦然。

功能特性

功能 G37 是非模态的，所以，每次执行切入时必须编写。



7.7 切出 (G38)

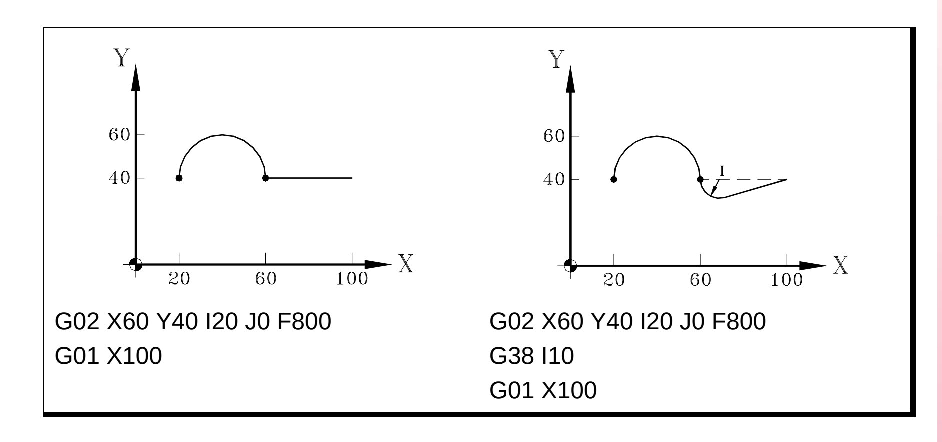
功能 G38 可以用于刀具切出结束加工，而不必计算交点。

编程

切出必须单段编写，该段前的路径被修改；路径必须是直线 (G00 或 G01)。

编程格式 "G38 I< 半径 >"，半径值单位为毫米或英尺，取决于当前的单位制。

切出后的直线路径长度必须等于或大于两倍的切入半径。同样地，半径必须是正的，并且当带刀具半径补偿工作时，其必须大于刀具半径。



注意事项

切出半径 "I" 值保持有效，直到编写另一个值，所以，不需要在连续的同尺寸的切出操作中编写。

倒角尺寸 "I" 值也用于功能：

G36 (圆角) 作圆弧半径 。

G37 (切入) 作切入半径。

G39 (倒角) 作倒角尺寸。

这意味着当编写其中一条指令时，在 G38 设置的切出半径将成为切入半径，圆弧半径或倒角尺寸，反之亦然。

功能特性

功能 G38 是非模态的，所以，每次执行切入时必须编写。



7.8 镜像 (G11, G12, G13, G10, G14)

镜像可以用于在相对于一根或多根轴的对称位置重复编写加工操作。当使用镜像时，镜像轴执行相反符号的运动。

编程

用以下功能可以应用镜像：

G10	镜像撤销
G11	相对于 X 轴镜像
G12	相对于 Y 轴镜像
G13	相对于 Z 轴镜像
G14	编程方向镜像

G10 镜像撤销

对所有轴撤销镜像，包括用 G14 激活的镜像。

如果添加路径定义程序段，在运动前镜像将被撤销。

G11 to G13 X 轴， Y 轴或 Z 轴镜像

功能 G11， G12 和 G13 激活分别相对于 X， Y 和 Z 轴的镜像。这些功能不互相取消，因而，可以同时对数根轴使用镜像。

如果没有添加路径定义程序段，在运动前镜像将有效。

```
G11
   (相对于 X 轴镜像)
G12
   (相对于 Y 轴镜像，相对于 X 轴镜像保持有效)
...
G10
   (对所有轴撤销镜像)
```



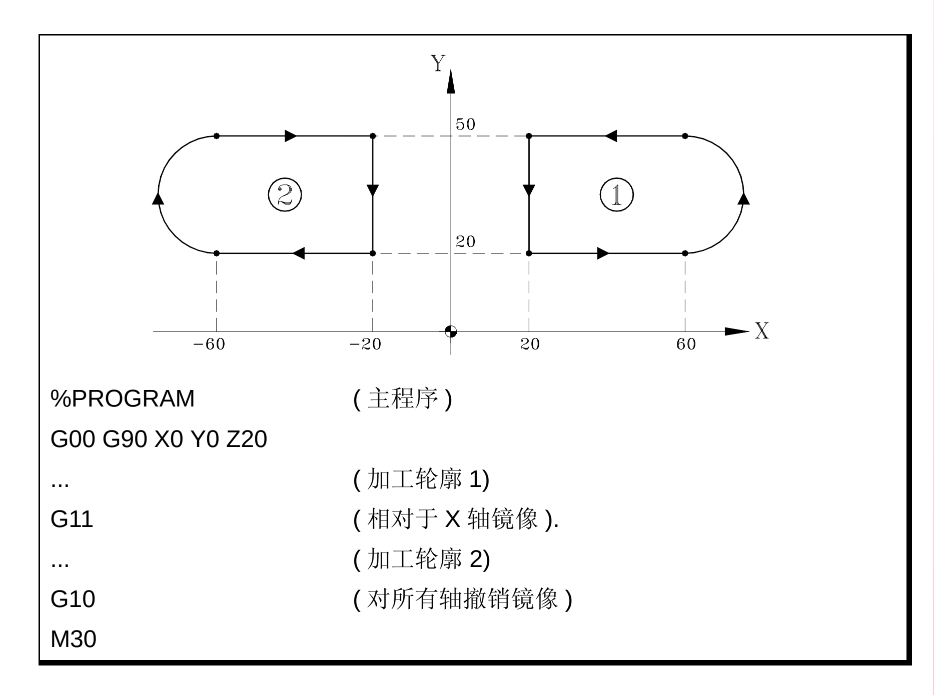

G14 编程方向镜像

可以用于在任何轴激活或撤消镜像。编写功能 G14 定义激活或撤销镜像，然后轴跟定义该轴是激活 (< 轴 >=-1) 还是撤销 (< 轴 >=1) 镜像的值。

```
G14 X-1 V-1
    ( 相对于 X 和 V 轴镜像 )
G14 X1
    ( 撤销相对于 X 轴镜像，保持相对于 V 轴镜像 )
...
G14 V1
    ( 撤销相对于 V 轴镜像 )
```

注意事项

使用镜像加工轮廓时，加工方向和编程方向相反。如果轮廓定义时使用刀具半径补偿，当激活镜像时，数控系统将改变相对获得编程轮廓的刀具半径补偿的类型 (G41 或 G42) 。



%PROGRAM	(主程序)
G00 G90 X0 Y0 Z20	
...	(加工轮廓 1)
G11	(相对于 X 轴镜像).
...	(加工轮廓 2)
G10	(对所有轴撤销镜像)
M30	

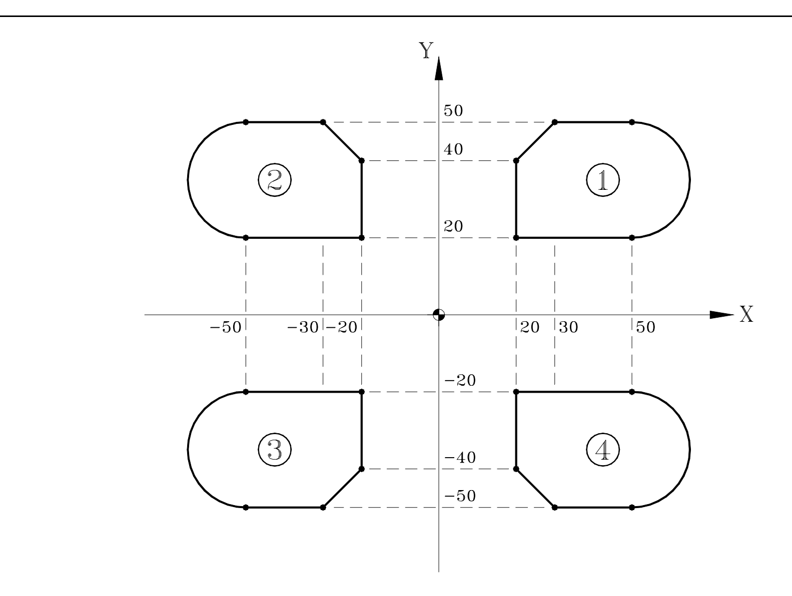
功能特性

功能 G11, G12, G13 和 G14 是模态的。一旦在一根轴上激活镜像，可以保持有效一直到用 G10 或 G14 撤销。

功能 G10 和 G14 是相互不兼容的，和 G11, G12 以及 G13 也是。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用 G10。






```
%L PROFILE                ("PROFILE" 子程序定义 )
 N10 G00 X10 Y10
 N20 G01 Z0 F400
 N30 G01 X20 Y20 F850
 N40 X50
 N50 G03 X50 Y50 R15
 N60 G01 X30
 N70 X20 Y40
 N80 Y20
 N90 X10 Y10
 N100 Z10 F400
M29                       ( 子程序结束 )

%PROGRAM                  ( 主程序 )
 N10 G0 X0 Y0 Z10
 N20 LL PROFILE           ( 调用子程序，加工轮廓 1)
 N30 G11                  ( 相对于 X 轴镜像 ).
 N40 LL PROFILE           ( 调用子程序，加工轮廓 2)
 N50 G12                  ( 相对于 X 和 Y 轴镜像 ).
 N60 LL PROFILE           ( 调用子程序，加工轮廓 3)
 N70 G14 X1               ( 撤销相对于 X 轴镜像 )
 N80 LL PROFILE           ( 调用子程序，加工轮廓 4)
 N90 G10                  ( 对所有轴撤销镜像 )
 N100 G00 X0 Y0 Z50
M30
```

7.9 坐标系旋转，图形旋转 (G73)

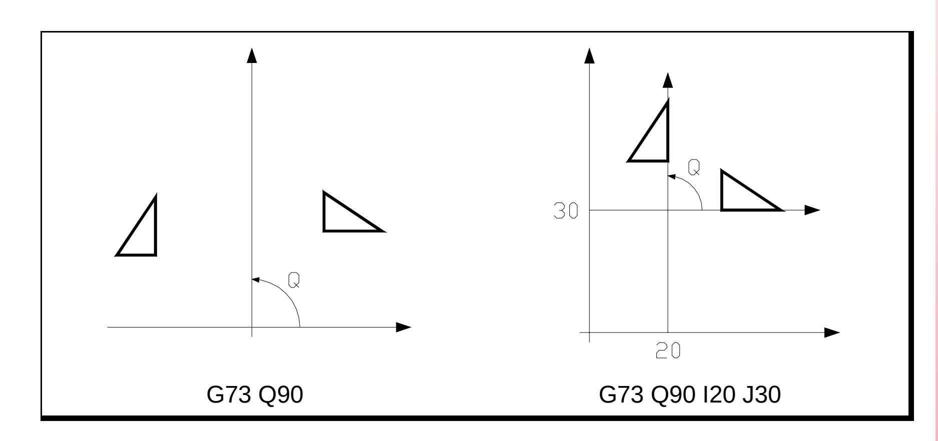
功能 G73 可以用于旋转坐标系，将当前的坐标参考系原点（工件零点）作为旋转中心或编写旋转中心。

编程

坐标系旋转必须单段编写。编程格式是 "G73 Q I J"，其中 ：

Q	指明旋转的角度，单位度。
I, J	定义旋转中心的横坐标和纵坐标。必须用绝对坐标相对于工件零点编写。 编程时，两个参数必须都编写。 如果不编写，工件零点将作为旋转中心。

撤销坐标 （图形）旋转，编写功能 G73，单段编写，无其它数据。

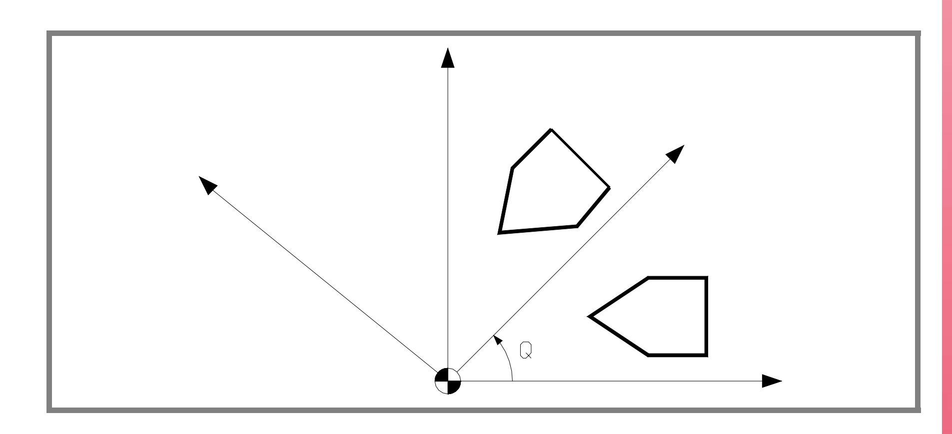


所以，功能 G73 可以如下所示编程：

G73 Q I J	旋转角度 "Q"，相对于工件零点的横坐标 "I" 和纵坐标 "J" 。
G73 Q	旋转角度 "Q"，工件零点作为旋转中心 。
G73	撤销坐标 （图形）旋转。

注意事项

功能 G73 是增量的； 也就是各种编写的 "Q" 值是累加的。



"I" 和 "J" 值受当前有效镜像影响。如果任意镜像功能有效，数控系统先应用镜像，然后才是坐标旋转。



功能特性

功能G73是模态的，坐标旋转保持有效，一直到使用功能G73撤销或改变工作平面。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统撤销有效的坐标系（图形）旋转。

编程实例

假定初始点为 X0 Y0:

```
%L PROFILE           ( 轮廓子程序 )
 G01 X21 Y0 F300
 G02 Q0 I5 J0
 G03 Q0 I5 J0
 G03 Q180 I-10 J0
M29                  ( 子程序结束 )

%PROGRAM             ( 主程序 )
$FOR P0=1, 8, 1      ( 重复加工轮廓，图形旋转 8 次 )
 LL PROFILE          ( 轮廓加工 )
 G73 Q45             ( 坐标旋转 )
$ENDFOR
M30
```




7.10 缩放因子

可以用于放大或缩小编程轮廓的尺寸。可以用一个程序制造一系列形状相似，大小不同的零件。

缩放因子应用于该通道的所有轴。在激活缩放因子后，所有的编程坐标将用定义的缩放因子增加一直到撤销或编写新的缩放因子。

激活缩放因子

该缩放因子可以用指令 G72 或 #SCALE 激活。可用其中任何一条。

虽然有两条不同的指令，缩放因子相同；也就是用 G72 编写的缩放因子可修改用 #SCALE 编写的缩放因子，反之亦然。

用 G72 编程

编写功能 G72，然后 是由参数 S 设置的缩放因子，如下所示。

G72 S< 缩放因子 >

单独编写功能 G72 或编写缩放因子 1，撤销当前的缩放因子。

参数 "S" 设置缩放因子必须编写在功能 G72 之后。如果编写在前面，将被认为是主轴速度。

用 #SCALE 编程

编写指令 #SCALE，然后是缩放因子，如下所示 。必须编写括号。

#SCALE [< 缩放因子 >]

编写缩放因子 1，撤销当前的缩放因子。

```
#G72 S2
#SCALE [3]
#G72
#SCALE [1]
```

撤销缩放因子

普通缩放比例因子撤销，用功能 G72 或 #SCALE，设置缩放因子为 1。

当用功能 G72，单段编写 G72 撤销缩放因子。



注意事项

激活机床坐标系 (#MCS ON) 可临时撤销缩放因子，直到撤销机床坐标系 (#MCS OFF)。

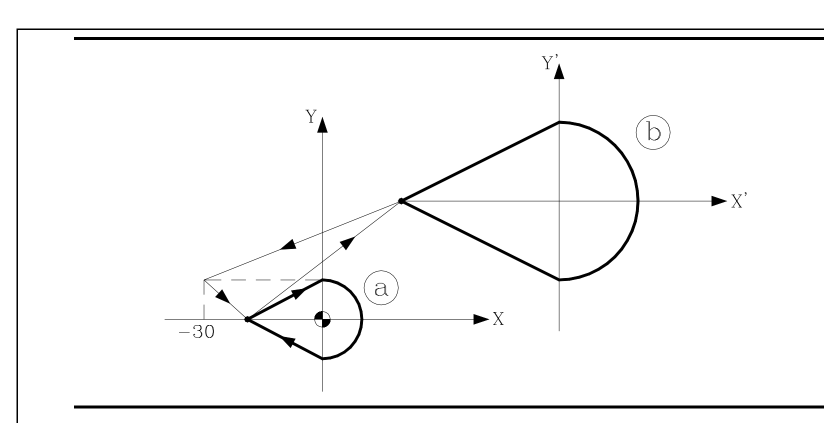
当机床坐标系有效时，缩放因子不能被激活或修改。

功能特性

缩放比例保持有效，直到用另一个缩放因子撤销。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统撤销缩放因子。

编程实例



%L PROFILE	(待加工轮廓)
G90 X-19 Y0	
G01 X0 Y10 F150	
G02 X0 Y-10 I0 J-10	
G01 X-19 Y0	
M29	
%PROGRAM	
G00 X-30 Y10	
#CALL PROFILE	(加工轮廓 "a")
G92 X-79 Y-30	(坐标预置)
#SCALE [2]	(施加缩放因子 2)
#CALL PROFILE	(加工轮廓 "b")
#SCALE [1]	(撤销缩放因子)
M30	



附加准备功能

8

8.1 暂停 (G04)

该指令可用于在特定时刻中断程序执行。

编程

该功能可以用指令 G04 或 #TIME 激活。可以用下述两条指令的任一条。

用 G04 编程

编写功能 G04，然后用参数 K 设置时间，单位为秒，如下所示。

```
G04 K<时间>
```

```
G04 K0.5                (0.5 秒暂停)
P1=3
G04 KP1                 (3 秒暂停)
G04 K[P1+7]             (10 秒暂停)
```

当用常数编写时间时，还允许使用如下格式。

```
G04<时间>
```

```
G04 5                   (5 秒暂停)
```

用 #TIME 编程

编写指令 #TIME，然后是时间，单位为秒，如下所示。括号为可选项。

```
#TIME [<时间>]
```

```
#TIME [5]               (5 秒暂停)
P1=2
#TIME P1                (2 秒暂停)
```

功能特性

功能 G04 是非模态的，所以，需要暂停时必须每次编写 。

功能 G04 也可以编写成 G4。

8.2 软行程极限编程 (G198-G199)

每根轴的软行程极限可以通过以下功能来改变：

G198	设置软行程下限
G199	设置软行程上限

当编写 G198 或 G199 时，数控系统认为接下来编写的坐标系设置新的软行程极限 。

```
G198 X-1000 Y-1000
    ( 新下限 X=-1000 Y=-1000)
G199 X1000 Y1000
    ( 新上限 X=1000 Y=1000)
```

依据有效的 G90 或 G91 模式，新极限位置定义用机床参考系的绝对坐标，或相对于当前有效极限位置的增量坐标。

```
G90
G198 X-800
    ( 新下限 X=-800)
G199 X500
    ( 新上限 X=500)
G90 X-800
G91
G198 X-700
    ( 新增量下限 X=-1500)
```

注意事项

用 0 值定义两个行程极限撤销该轴的行程极限，包括用参数设置的。要恢复行程极限，必须再次编写。

两个行程极限可以是正值或负值；但是下行程极限必须小于上行程极限。否则，可能在任何方向都不能运动。

如果在设置新的行程极限后，某根轴的位置越过了行程极限，轴可能向工作区 （两极限之间）运动。

功能特性

功能 G198 和 G199 是模态的，互不兼容。

上电或确认机床参数后，数控系统采用机床参数设置的软行程极限。

执行 M02 或 M30 或在急停或复位后，数控系统保持 G198 和 G199 设置的软行程极限。

8.3 Hirth 轴 (G170-G171)

Hirth 轴 可以通过编程撤销或激活。当 Hirth 轴有效时，它仅能到达具体位置，然而，当它被取消时， 工作方式像普通旋转轴或直线轴一样，能到达任何位置。

编程

Hirth 轴可用以下功能撤销或激活 :

G170	Hirth 轴取消
G171	Hirth 轴激活

激活或取消 Hirth 轴，编写相应功能，然后轴被激活或取消，数字表示轴被激活的顺序。

采取设置 B 轴和 C 轴为 旋转 Hirth 轴，间距为 10º 。

```
G171 B1 C2          ( 激活 B 和 C 轴作 Hirth 轴 )
G01 B50 C20         ( 插补两根轴 )
...
G170 B1             (B 轴取消 )
G01 X100 B33
```

如果激活 Hirth 轴时，它定位在错误的位置，数控系统将发出警报，这样操作者可以旋转它到正确的位置。

注意事项

Hirth 轴必须始终定位在特定的位置。当定位时，考虑有效零点偏置（预置或零点偏置）。

直线轴和旋转轴都可以是 Hirth 轴。只能激活由机床参数 [A.M.P. "HIRTH"] 定义为 Hirth 轴的轴才可以激活为 Hirth 轴。

功能特性

功能 G170 和 G171 是模态的，互不兼容。

上电后，执行 M02 或 M30 或在急停或复位后，数控系统激活所有的 Hirth 轴。

8.4 OEM 子程序 (G180-G189)

OEM 子程序由机床厂商定义。机床厂商可以定义最多 10 个子程序，并和功能 G180 到 G189 关联。当执行这些功能时，也会执行相应的子程序。

执行和功能关联的子程序时，产生新的嵌套层 （最多 7 层嵌套）。

编程

功能 G180 到 G189 允许初始化子程序的局部参数。在调用功能的子程序后可以定义参数值，可以用参数 P0-P25 或字母 A-Z (除了 "Ñ"), "A" 对应 P0 和 "Z" 对应 P25。

```
%PROGRAM
   F1000
   P0=10 P1=20 P2=30
   G1 XP0 YP1 ZP2
   G180 P0=100 P1=200 C300    ( 初始化参数 )
M30
%SUB_180                      ( 与 G180 关联子程序 )
   G1 XP0 YP1 ZP2
M29
```

在主程序中，轴运动到 X10 Y20 Z30。执行子程序，轴运动到 X100 Y200 Z300。

除初始化参数之外，任何其它类型的附加信息可以添加到这些功能，甚至是运动。这些信息必须在调用功能子程序之前被编写；另外，这些数据将被认为是初始化参数。

一旦执行相关子程序，在程序段内，结束执行编写的其余信息。

执行编程运动，然后，与 G180 相关子程序，设置参数 P0 和 P1。

```
G01 X50 F450 G180 P0=15 P1=20
```

所有数据被认为是参数设置，其中 P6(G)=1, P23(X)=50 and P5(F)=450。

```
G180 P0=15 P1=20 G01 X50 F450
```

注意事项

主程序（或子程序）可以调用子程序，子程序可以调用另一个子程序，依此类推。数控系统对子程序调用的限制是最大 20 层嵌套。当在子程序里使用局部参数，除产生新的嵌套以外，对局部参数也产生新的嵌套级别；在 20 层子程序嵌套中，可以有最多 7 层局部参数嵌套 。

功能特性

G180 到 G189 是非模态功能。

8.5 轴的参数范围的变化 (G112)

数控系统可以为每根轴设置 4 套参数，定义不同的动态参数（加速度，增益等）。

参数设置可以用功能 G112 编程来选择。该功能在机床上不执行任何物理变化（换档），只采用有效的参数。

当使用 Sercos 轴时，功能 G112 也包括变换驱动的速度档。

编程

轴参数范围变化。

要选用不同的参数组，编写 G112，然后，分别选择轴和每根轴的新参数组。

```
...
G112 X2 Y3    ( 为 X 选择第 2 组参数，为 Y 轴选择第 3 组参数 )
...
```

为主轴变换参数组数。

这种情况下，将在点动模式 (M19) 下改变参数组。当工作在速度模式 (M03/M04) 下时，功能 G112 仅改变参数组；不同于 M41 到 M44，不产生物理换档。

```
...
G112 S2       ( 选择第三组主轴参数 )
...
```

当用 M41 到 M44 换挡时，不需要编写 G112。

功能特性

功能 G112 是模态的。

在确认机床参数之后，每次执行一个程序是从自动模式开始，上电后，执行 M02 、M30 或急停或复位后，数控系统根据赋给机床参数 DEFAULTSET 的值运行。

如果 DEFAULTSET 不等于 0，保持 G112 定义的范围。否则，采取机床参数 DEFAULTSET 定义的范围。

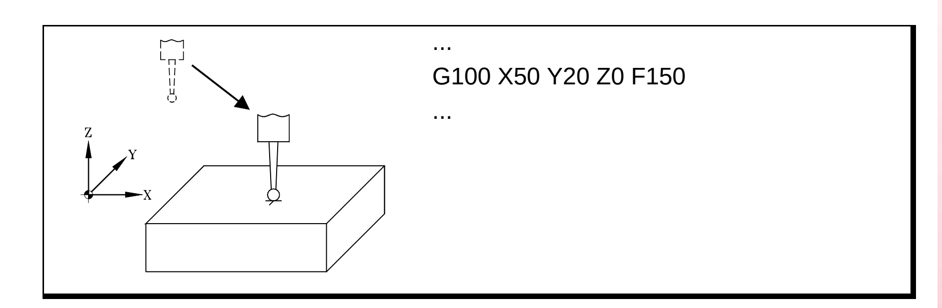
8.6 探测 (G100)

用功能 G100，可以编写运动，该运动在数控系统收到探针信号时结束。

操作

探针运动定义用功能 G100，后跟探针目标点坐标。

探针将沿编程路径运动，直到数控系统收到从探针来的信号或到达编程位置。在那点，程序段完成，并且数控系统采用当前轴位置作为理论位置。



如果数控系统在到达编程目标点之前收到探针信号，数控系统将使用 G101 采用轴的理论位置。See *"8.6.1 包含／排除探针偏置 (G101/G102)"* on page 140.

进给率运行方式

探针进给率是有效的 "F"，进给率受每个探针轴的机床参数 PROBEFEED 的限制 。这个值也被参数 PROBERANGE 和 PROBEDELAY 限制，所以如果加速度和轴的骤起停有效，将始终不会妨碍最大探测距离。

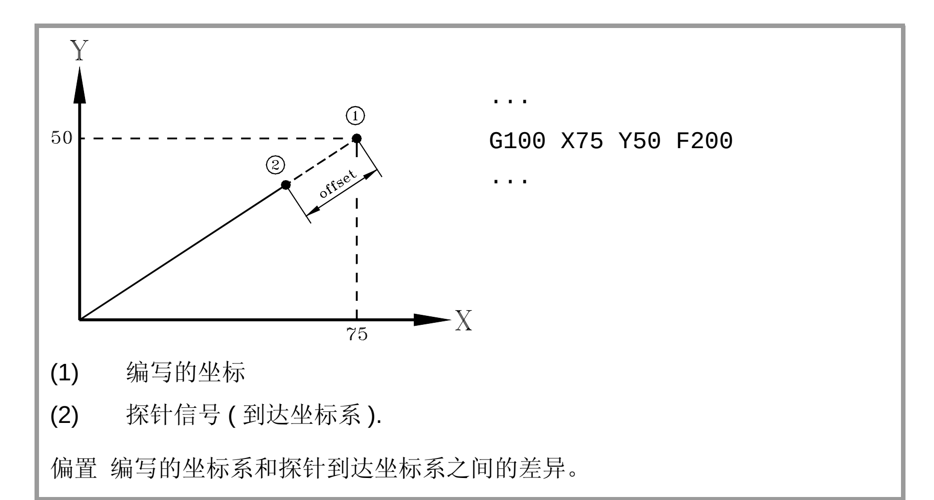
可通过数控操作面板上的选择开关，或通过程序和 PLC，在 0% 到 200% 之间修调编写的进给率 "F"。然而，最大倍率受机床参数 MAXOVR 限制。

功能特性

功能 G100 是非模态的，所以只要需要新的探针运动时必须编写 。

8.6.1 包含 / 排除探针偏置 (G101/G102)

探针偏置在编写的坐标系和探针到达坐标系之间是不同的。



(1) 编写的坐标

(2) 探针信号 (到达坐标系).

偏置 编写的坐标系和探针到达坐标系之间的差异。

编程

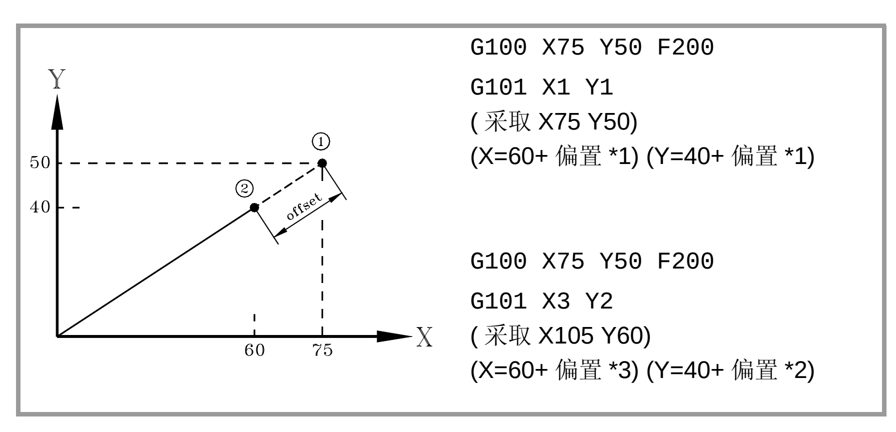
和探针偏置有关的功能:

G101	包含探针偏置
G102	排除探针偏置

G101 **包含由测量产生的偏置**

用此功能,数控系统将考虑探针偏置,设置理论轴位置;换句话说,数控系统采用编写的坐标作为理论轴位置 (探针到达点 + 探针偏置).

包含探针偏置,编写 G101,后跟含探针偏置的轴和每根轴要包含的因素。此因素表示包含偏置的次数。



功能 G101 仅能在探针后执行 。

G102 排斥由测量产生的偏置

使用这种功能，数控系统忽略探针偏置，设置理论轴位置。

排除探针偏置定义，编写功能 G102，后跟偏置受排斥的轴。

```
...
G102 X Y          ( 排斥 X 和 Y 轴的偏置 )
...
G102              ( 排斥所有轴的偏置 )
...
```

单独编写 G102 将撤销所有轴的偏置 。

功能特性

功能 G101 和 G102 是模态的，互不兼容。

上电后，执行 M02 或 M30 或在急停或复位后，数控系统保持 G101 编程值。 On

刀具补偿

9

刀具补偿允许直接编写零件轮廓和刀具半径，不考虑刀具尺寸。这样，不需要根据每把刀具的半径和长度，计算和重新定义刀具路径。

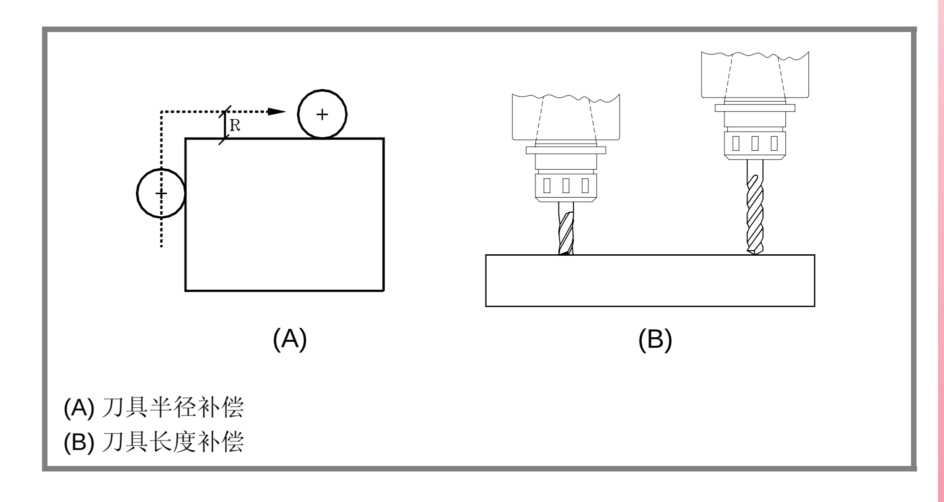
补偿类型

刀具半径补偿

使用刀具半径补偿时，刀具中心偏离编程轨迹一个等于刀具半径的距离。这样来获得编程零件的正确尺寸。

刀具长度补偿

使用刀具长度补偿时，数控系统补偿不同编程刀具之间的长度差。



(A) 刀具半径补偿
(B) 刀具长度补偿

补偿值

每次应用的补偿值从刀具尺寸计算。

- 刀具半径补偿，应用值是所选刀具半径和半径磨损的总和。
- 刀具长度补偿，应用值是所选刀具长度和长度磨损的总和。

刀具 "T" 和刀具偏置 "D"，包含刀具尺寸，可以在程序的任何位置选择，甚至在刀具补偿有效时 。如果没有选择刀具偏置，数控系统采取刀具偏置 "D1"。

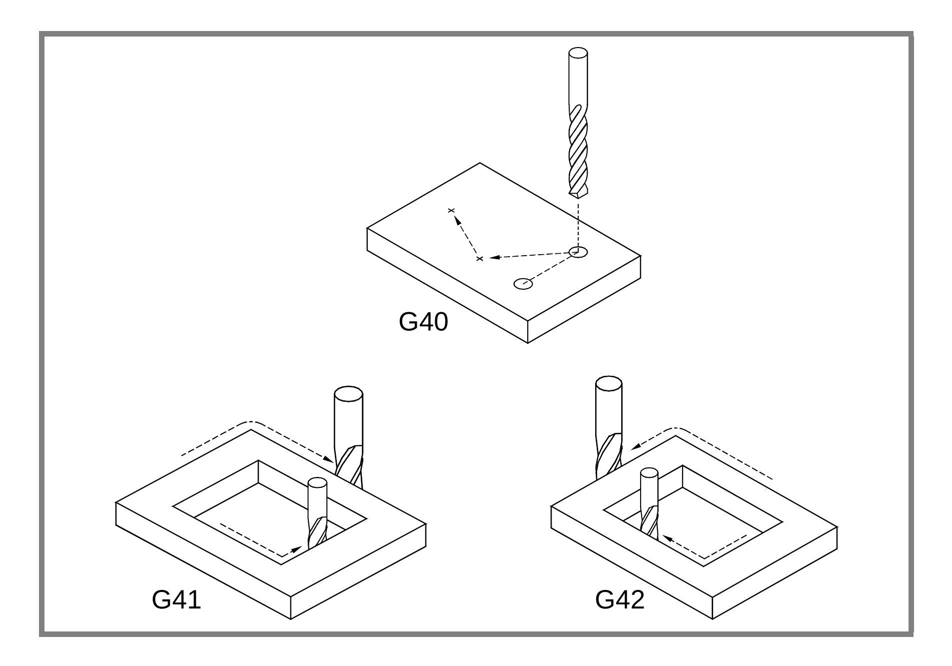
9.1 刀具半径补偿

先用功能 G17 (XY 平面), G18 (ZX 平面), G19 (YZ 平面) 或 G20 (用户自定义平面) 选择平面，在有效平面应用刀具半径补偿。

编程

选择刀具半径补偿的功能 :

G41	刀具半径左补偿
G42	刀具半径右补偿
G40	撤销刀具半径补偿



根据所选补偿类型(G41/G42)，刀具被放置在沿加工方向编程路径的左侧还是右侧，并且偏离编程路径距离等于刀具半径。如果没有选择刀具半径补偿 (G40)，数控系统把刀具中心置于编写的刀具路径上。

刀具半径补偿有效时，数控系统预先分析要执行的程序段，以便发现跟每一步有关的补偿错误、无效圆弧，等等。当发现时，数控系统不执行该程序段，即导致问题的程序段，并且屏幕显示警告，以提示操作者编程轮廓被修改了。对于每个轮廓修正，提出警告。

功能特性

功能 G40,G41 和 G42 是模态的，相互不兼容 。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G40。

9.1.1 与半径补偿相关的功能

与刀具半径补偿相关的功能可以编写在程序的任何位置，甚至当刀具半径补偿有效时。

选择程序段间过渡类型

程序段间过渡决定如何加入补偿路径。

编程

过渡类型可以通过编程选择，使用以下功能：

G136　　程序段间圆弧过渡

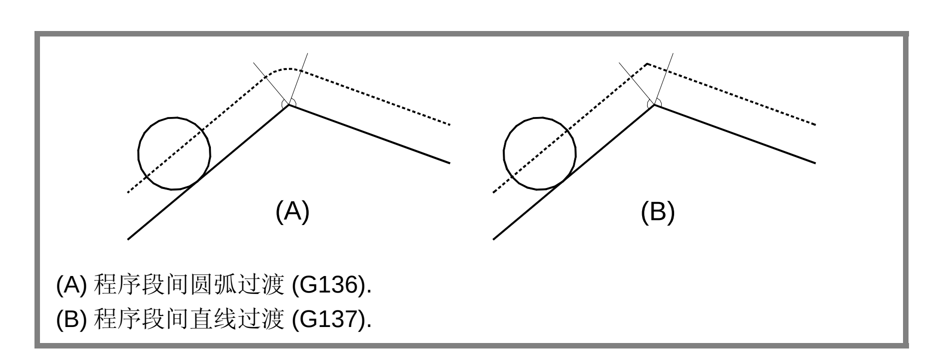
G137　　程序段间直线过渡

G136　**程序段间圆弧过渡**

功能 G136 有效，数控系统用圆弧路径加入补偿路径。

G137　**程序段间直线过渡**

功能 G137 有效，数控系统用直线路径加入补偿路径。



(A) 程序段间圆弧过渡 (G136).
(B) 程序段间直线过渡 (G137).

注释

这一章稍后的部分提供一张图表，介绍依据选择的不同过渡类型 (G136/G137)，如何加入不同路径。

功能特性

功能 G136 和 G137 是模态的，相互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用功能 G136 还是 G137，依据机床参数 IRCOMP 的设定值。

刀具半径如何激活和撤销

该功能与激活和撤销的策略有关，它确定刀具半径补偿如何开始和结束。

编程

策略类型可通过编程选择，用到如下功能：

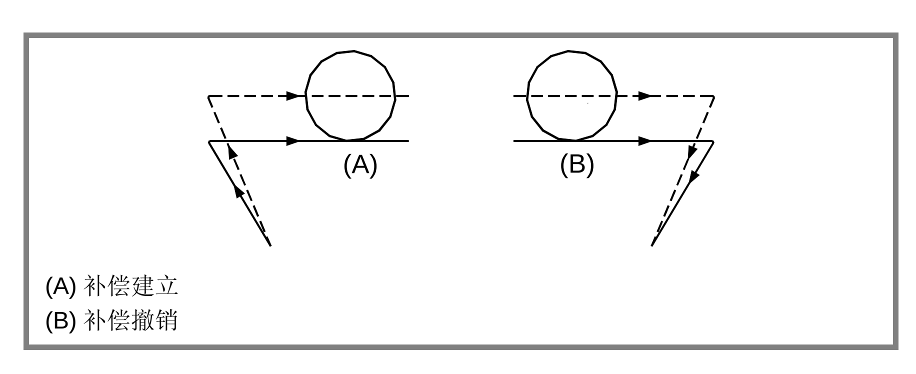
G138　　直接激活 / 撤销刀具补偿

G139　　间接激活 / 撤销刀具补偿

G138　直接激活 / 撤销刀具半径补偿

当补偿建立时，刀具直接运动至下一路径的垂直方向的位置 （无轮廓过渡）。

当补偿撤销时，刀具直接运动到编程终点 (无轮廓过渡)。

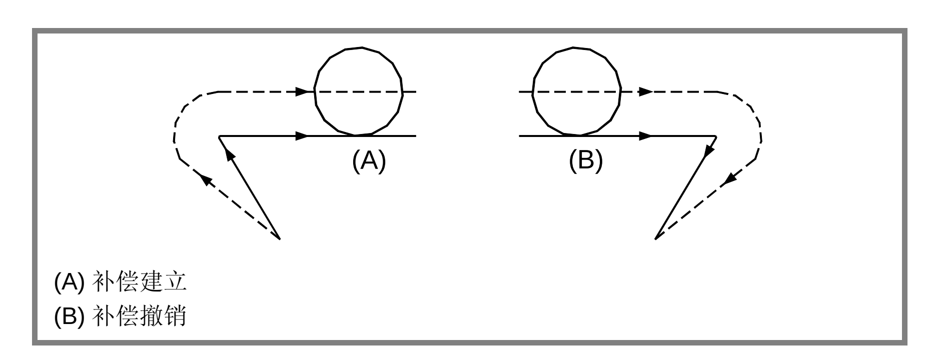


(A) 补偿建立
(B) 补偿撤销

G139　间接激活 / 撤销刀具半径补偿

当补偿开始时，刀具运动至下一路径的垂直方向的位置，有轮廓过渡。

当补偿撤销时，刀具运动到编程终点，有轮廓过渡。



(A) 补偿建立
(B) 补偿撤销

刀具在拐角处的路径取决于选择的转接类型 (G136/G37)。

注释

这一章稍后的部分提供一张图表，介绍依据选择的不同补偿类型 (G138/G139)，如何建立和撤销刀具半径补偿。

功能特性

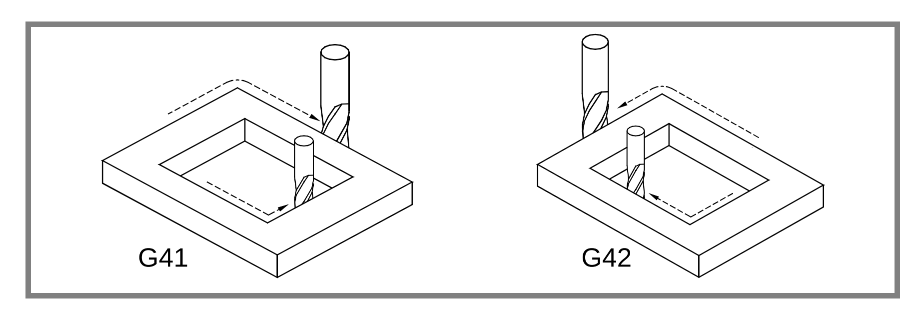
功能 G138 和 G139 是模态的，相互不兼容。

上电，执行 M02 或 M30 以及在急停或复位后，数控系统采用机床参数 IRCOMP 设定的功能。

9.1.2 刀具半径补偿建立

用以下功能选择刀具补偿：

G41	刀具半径左补偿。
G42	刀具半径右补偿。



执行其中的一个功能之后，刀具半径补偿在工作平面上的下一个运动开始有效，必须是直线运动。

刀具补偿路径如何开始，取决于 G138/G139 以及过渡类型 G136/G137 的选择：

- G139/G136

  刀具运动到下一路径的垂直方向，沿圆弧路径轮廓过渡。

- G139/G137

  刀具运动到下一路径的垂直方向，沿直线路径轮廓过渡。

- G138

  刀具直接运动到下一路径的垂直方向。不管编写哪种过渡类型 (G136/G137) 。

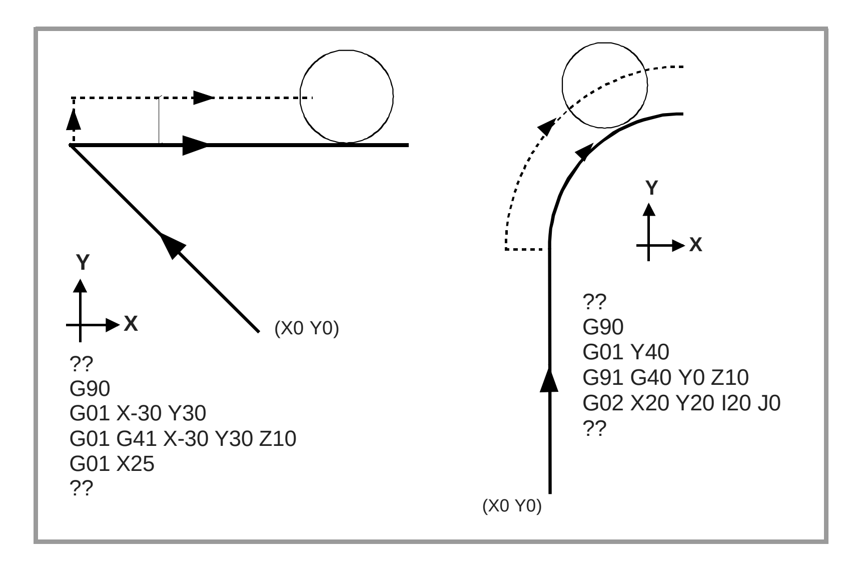
下面各页显示不同的刀具半径补偿开始的情况。其中，编程路径用实线表示，补偿路径用虚线表示。

无编程运动的补偿的建立

激活补偿之后，平面的轴有可能不包括在第一运动程序段中。例如，因为没有编写它们，或是当前刀具位置已经编程，或是编写了增量运动。

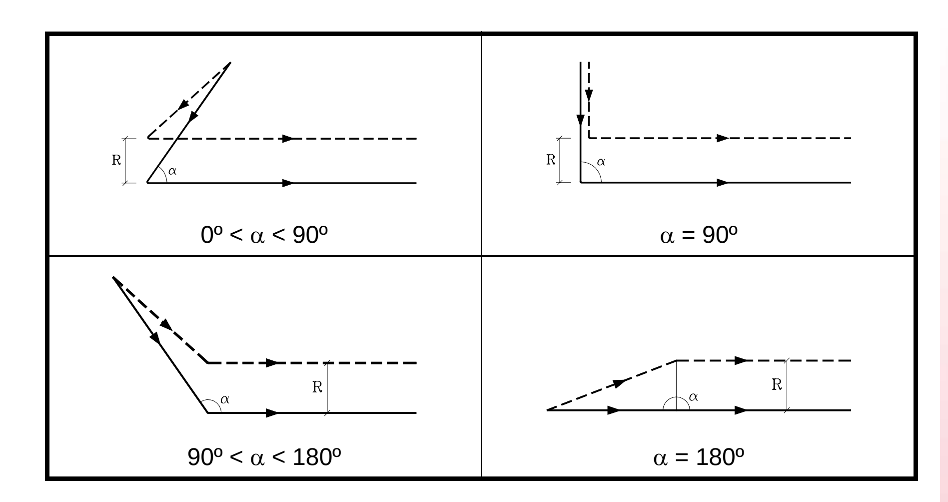
这种情况下，应用补偿在刀具所在的同一点，如下。根据在平面上的第一个的运动，刀具运动到路径起点的垂直方向的位置上。

编写的平面上的第一个运动可以是直线或圆弧。

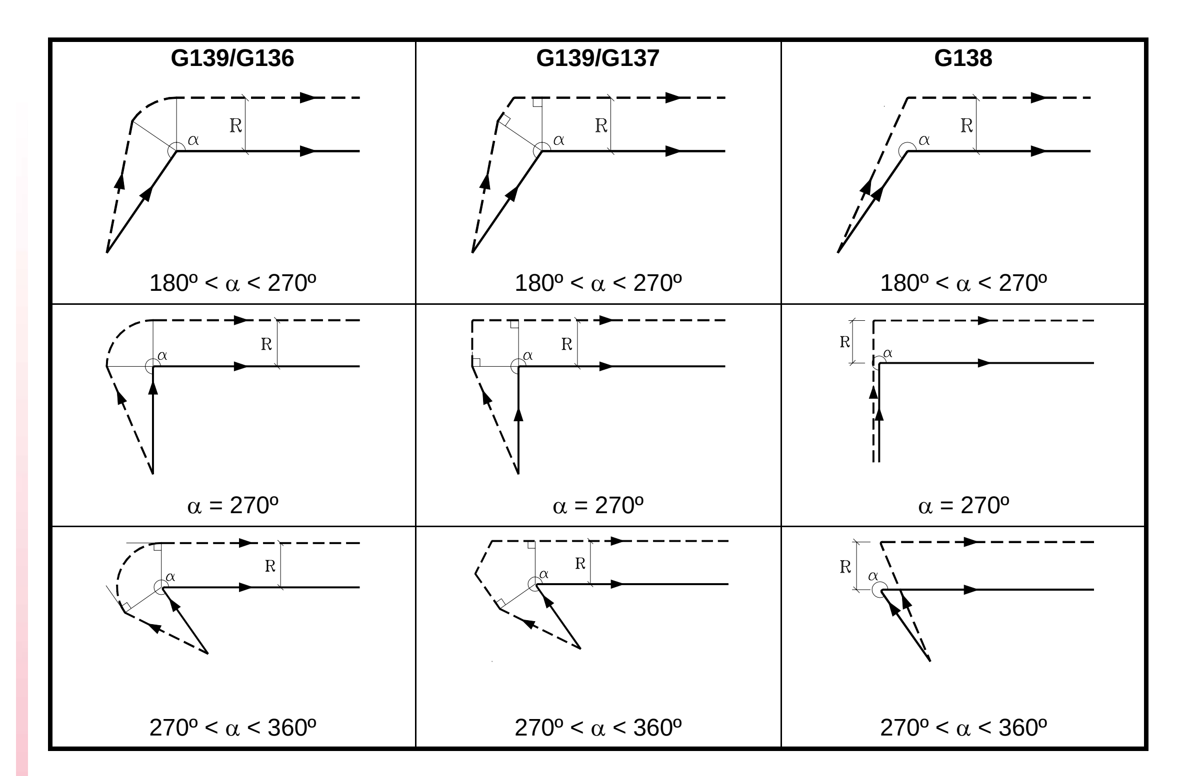


直线 -- 直线路径

当两路径间的夹角小于或等于180º时，刀具半径补偿建立路经不受功能G136/G137 或 G138/G139 选择的约束。

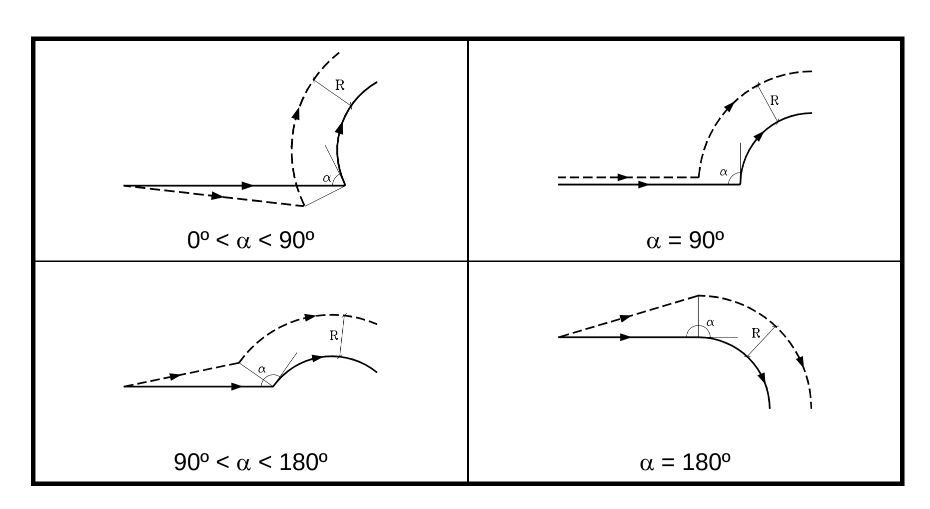


当两路径间的夹角大于 180º 时，刀具半径补偿建立路径受开始类型的功能选择（G138/G139）和过渡类型的选择（G136/G137）的约束。

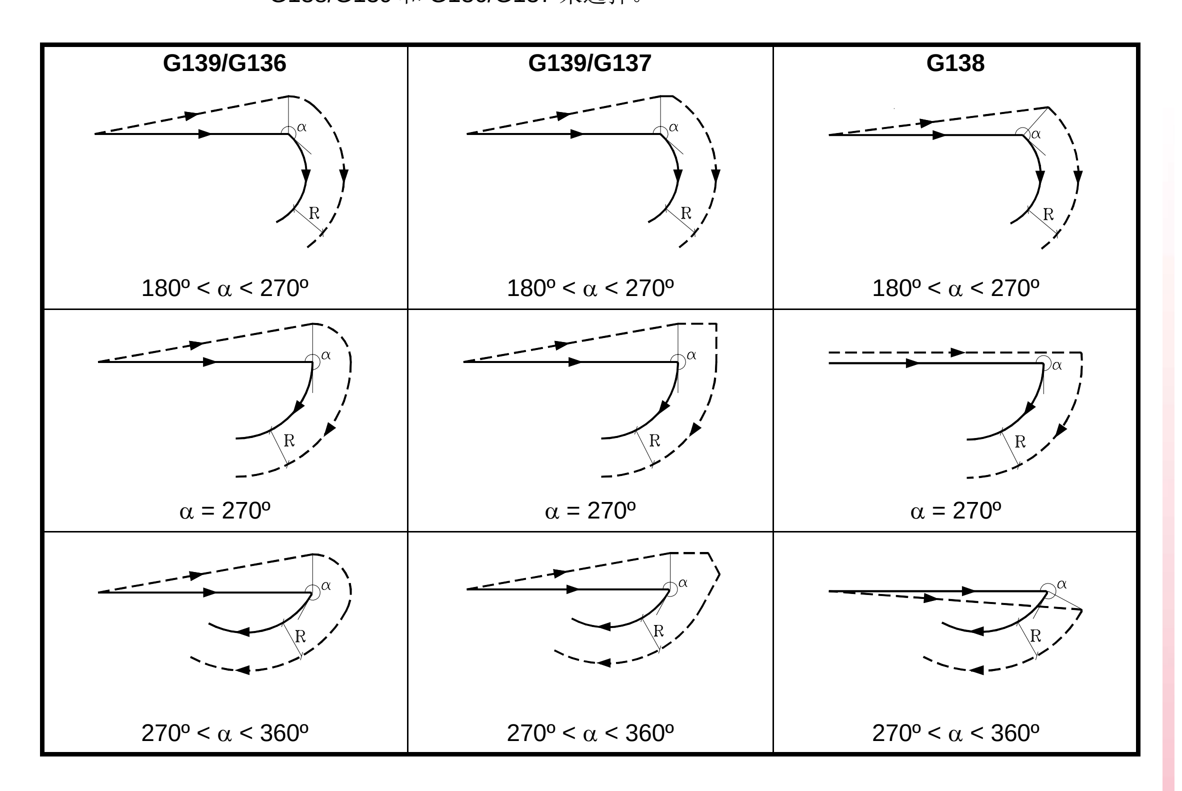


直线 -- 圆弧路径

当直线路径和圆弧切线的夹角小于或等于 180º 时，刀具半径补偿建立路径不受功能 G138/G139 和 G136/G137 的选择的约束。



当直线路径和圆弧切线的夹角大于 180º 时，刀具半径补偿建立路径根据功能 G138/G139 和 G136/G137 来选择。



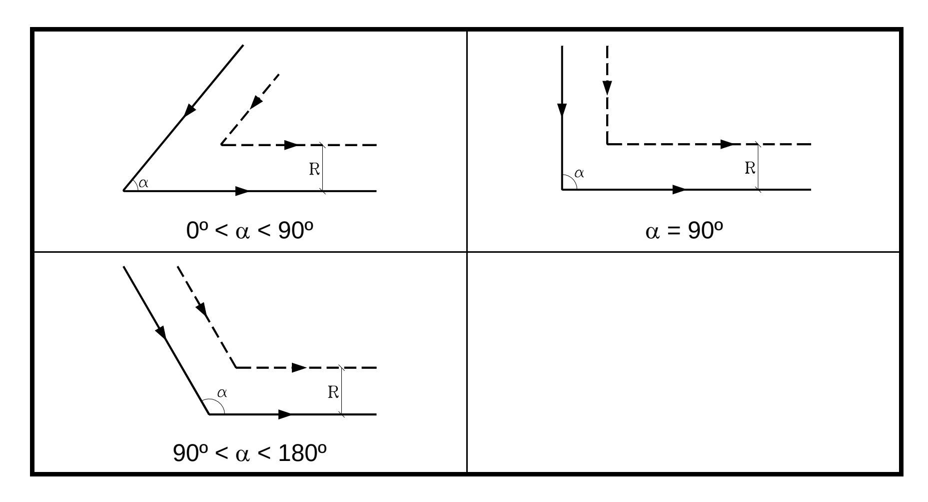
9.1.3 刀具半径补偿选择

补偿路径的加入方式取决于选择的过渡类型 (G136/G137)。

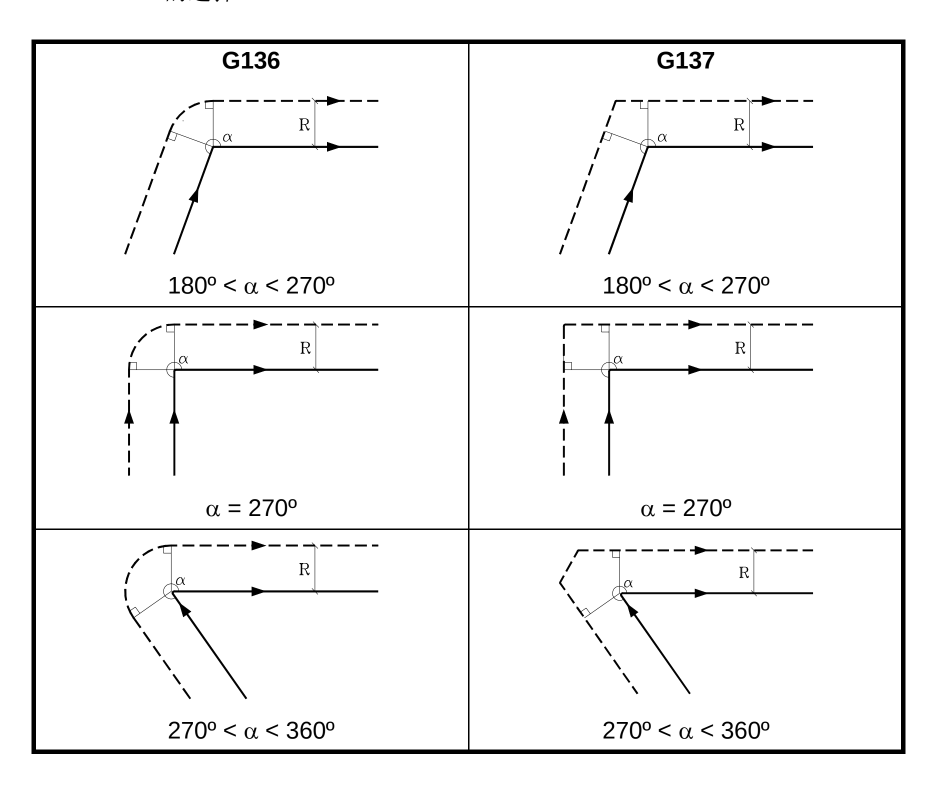
下列表格说明在各种路径之间可能的不同过渡方式，取决于功能 G136 或 G137 的选择。其中，编程路径用实线表示，补偿后的路径用虚线表示。

直线 -- 直线路径

当两路径之间的夹角小于或等于 180º 时，带刀具半径补偿的刀具路径的过渡与选择的功能 G136/G137 无关。

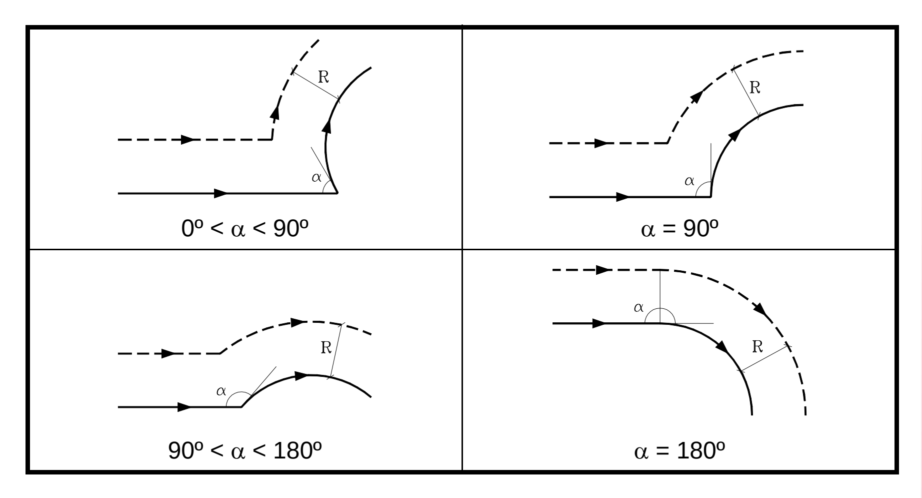


当两路径之间的夹角大于 180º 时，两补偿路径之间的过渡类型取决于功能 G136/G137 的选择 。

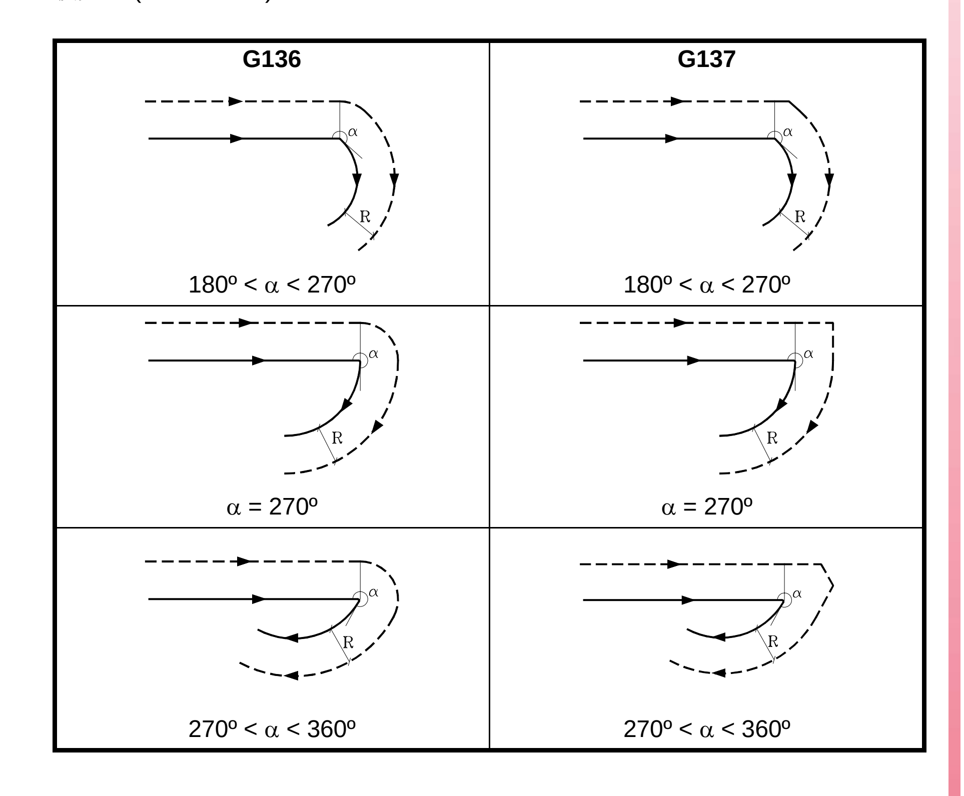


直线 -- 圆弧路径

当直线和圆弧切线之间的夹角小于或等于 180º 时，带刀具半径补偿的刀具路径的过渡与选择的功能 G136/G137 无关。

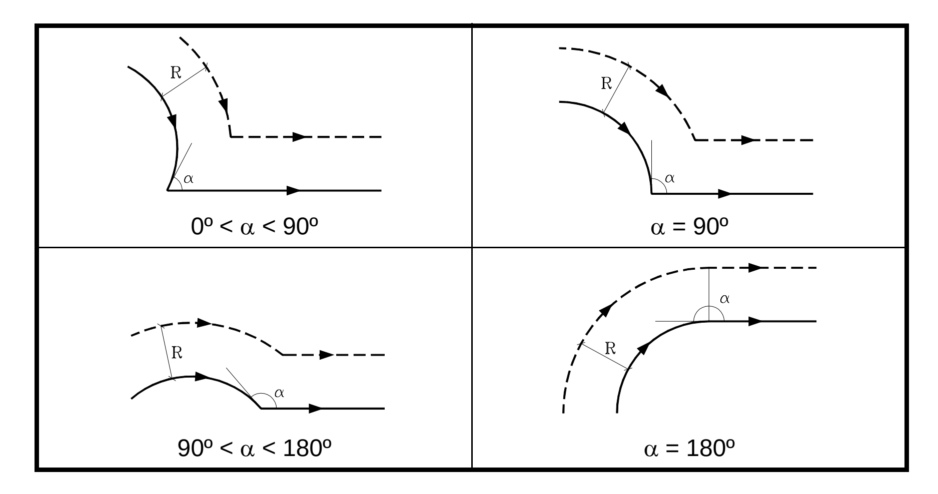


当直线路径和圆弧切线之间的夹角大于 180º 时，补偿路径的加入方式取决于所选过渡类型 (G136/G137)。

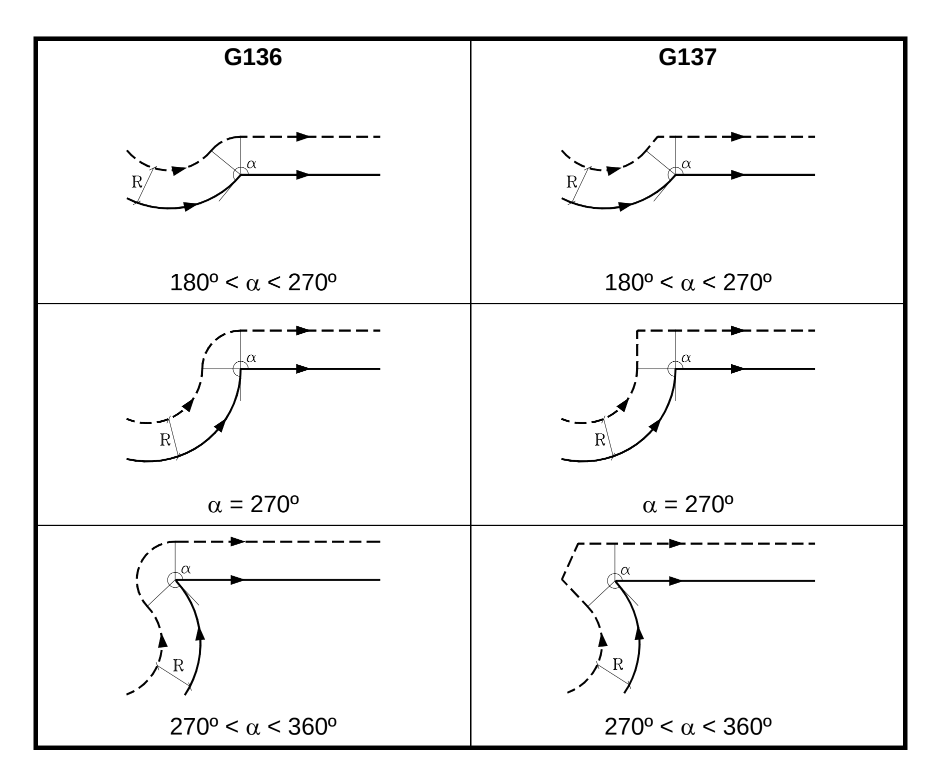


圆弧 -- 直线路径

当圆弧切线和直线的夹角小于或等于 180º 时时，带刀具半径补偿的刀具路径的过渡与选择的功能 G136/G137 无关。

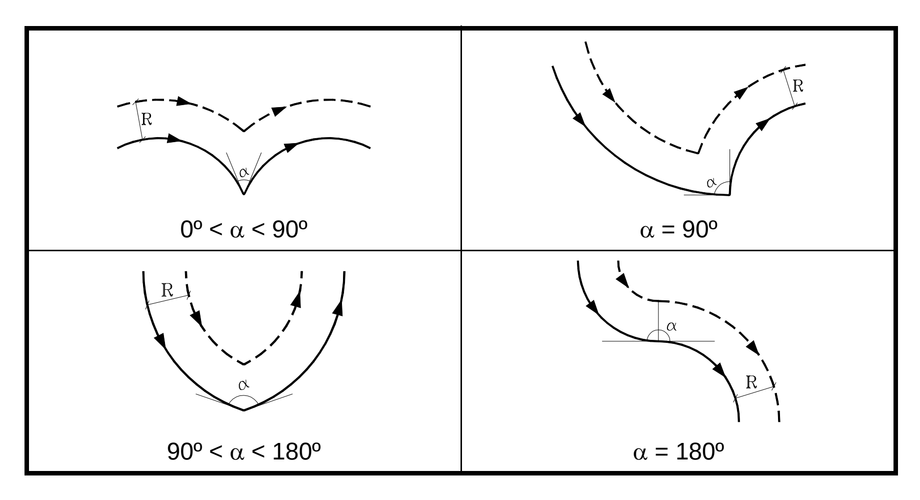


当圆弧切线和直线之间的夹角大于 180º 时，带刀具半径补偿的刀具路径的过渡根据选择的功能 G136/G137 来选择。

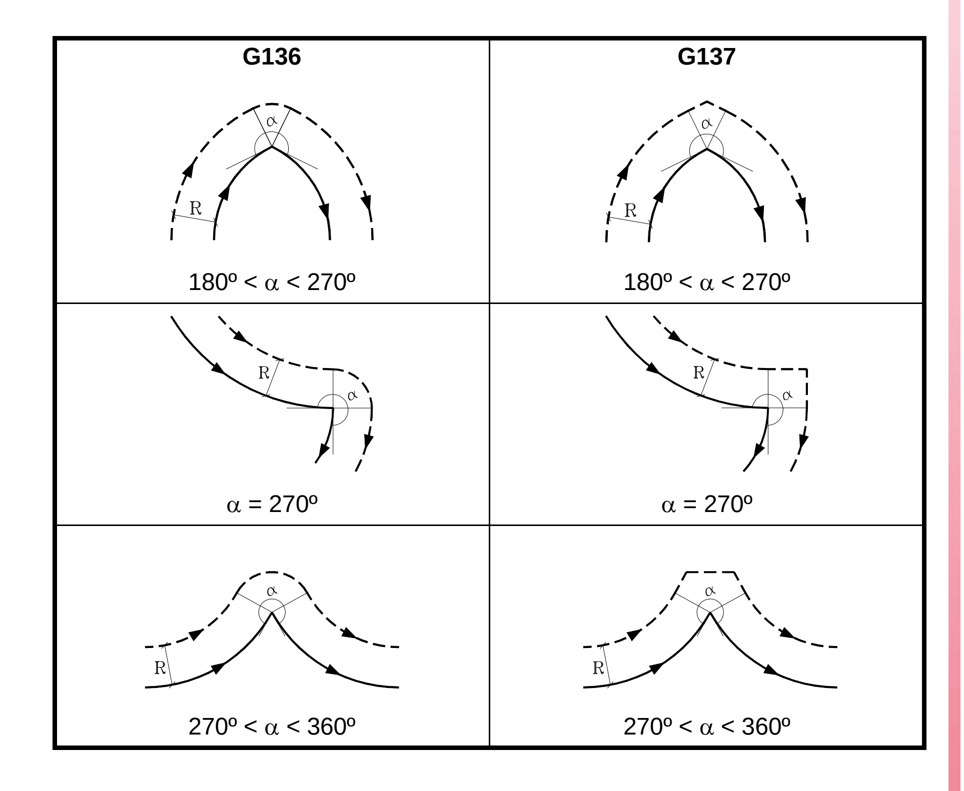


圆弧 -- 圆弧路径

当圆弧切线和圆弧切线的夹角小于或等于 180º 时，带刀具半径补偿的刀具路径的过渡与选择的功能 G136/G137 无关。



当圆弧切线和圆弧切线之间的夹角大于 180º 时，带刀具半径补偿的刀具路径的过渡根据选择的功能 G136/G137 来选择。



9.1.4 加工时刀具半径补偿变换

刀具半径补偿可从 G41 转换到 G42 或反之，而不需使用 G40 撤销。可以在任何运动程序段或甚至是无运动的程序段内转换；也就是，没有平面上轴的运动或编写相同的点两次。

转换前的最后一次运动和转换后的第一次运动被独立地补偿。转换刀具补偿类型，依照如下标准解决不同情况：

A. 补偿路径相互交叉。

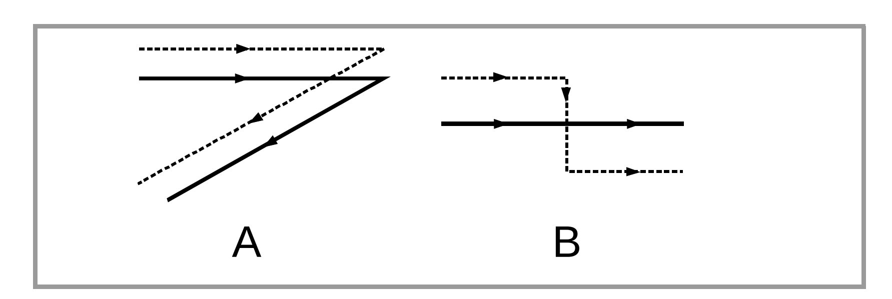
编程路径被补偿在各自对应的侧面。补偿侧面变换发生在两个路径的交叉点。

B. 补偿路径不相互交叉。

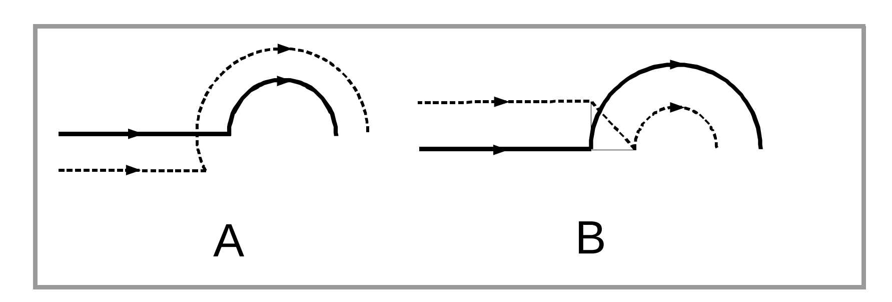
附加的部分嵌入两路径之间。从垂直点到终点的第一路径直到垂直点到起点的第二路径。两个点位于距编程路径 R 距离的位置。

不同情况总结：

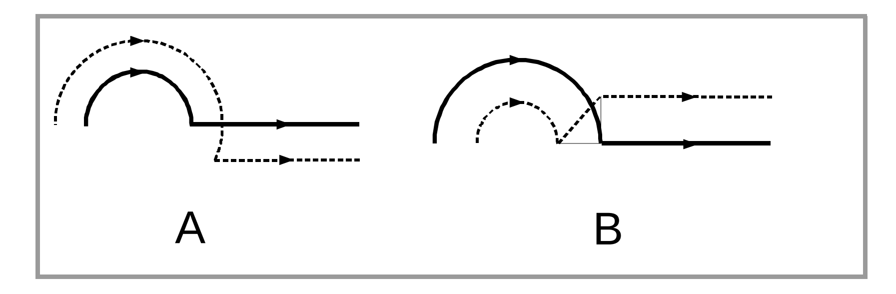
- 直线 - 直线路径：



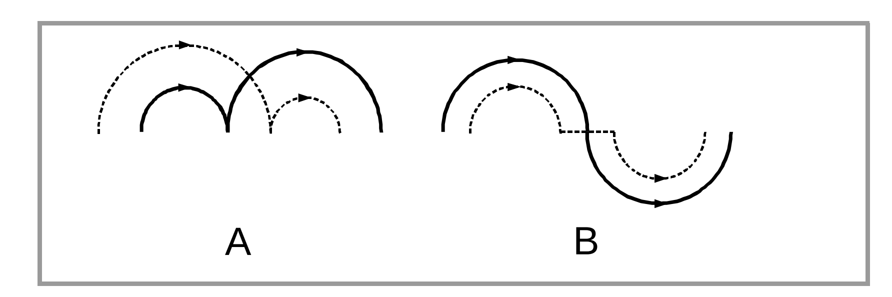
- 直线 - 圆弧路径：



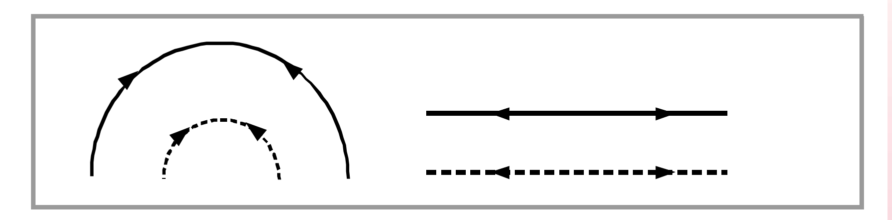
- 圆弧 - 直线路径：



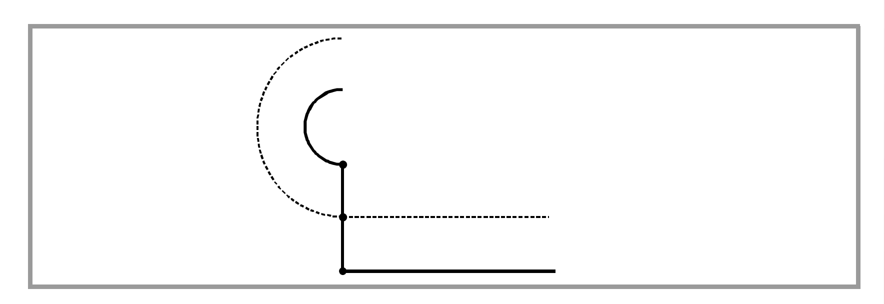
- 圆弧 - 圆弧路径：



- 沿同一路线往复路径。



- 与刀具半径一样长的中间路径：



9.1.5 刀具半径补偿撤销

刀具半径补偿用功能 G40 撤销。

在执行该功能中的一个之后，将在工作平面内的下一次运动时撤销刀具半径补偿，该运动必须是直线运动。

刀具半径补偿撤销路径依据撤销结束的类型 (G138/G139) 和过渡类型 G136/G137 来选择：

- G139/G136

  刀具运动到终点，沿圆弧路径做拐点等值线。

- G139/G137

  刀具运动到终点，沿直线路径做拐点等值线。

- G138

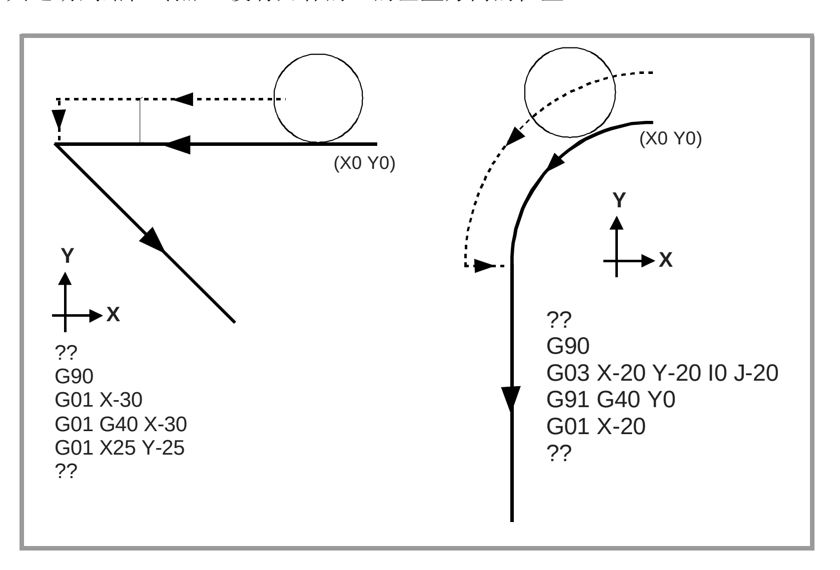
  刀具直接到终点，不管编写的过渡类型 (G136/G137) 。

下列表格说明不同情况下刀具半径补偿的取消路径。其中，编程路径用实线表示，补偿后的路径用虚线表示。

没有编程运动的刀具补偿撤销

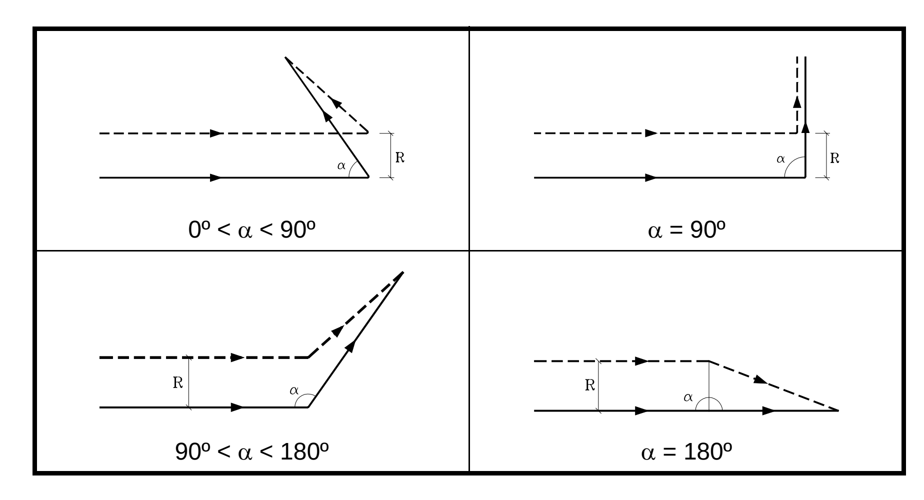
撤销刀补后，平面的轴有可能不包括在第一运动程序段中。例如，因为没有编写它们，或是当前刀具位置已经被编写，或是编写了增量运动。

这种情况下，应用补偿在刀具所在的同一点，如下。根据在平面上的第一个运动，刀具运动到路径终点 （没有刀补的）的垂直方向的位置上。

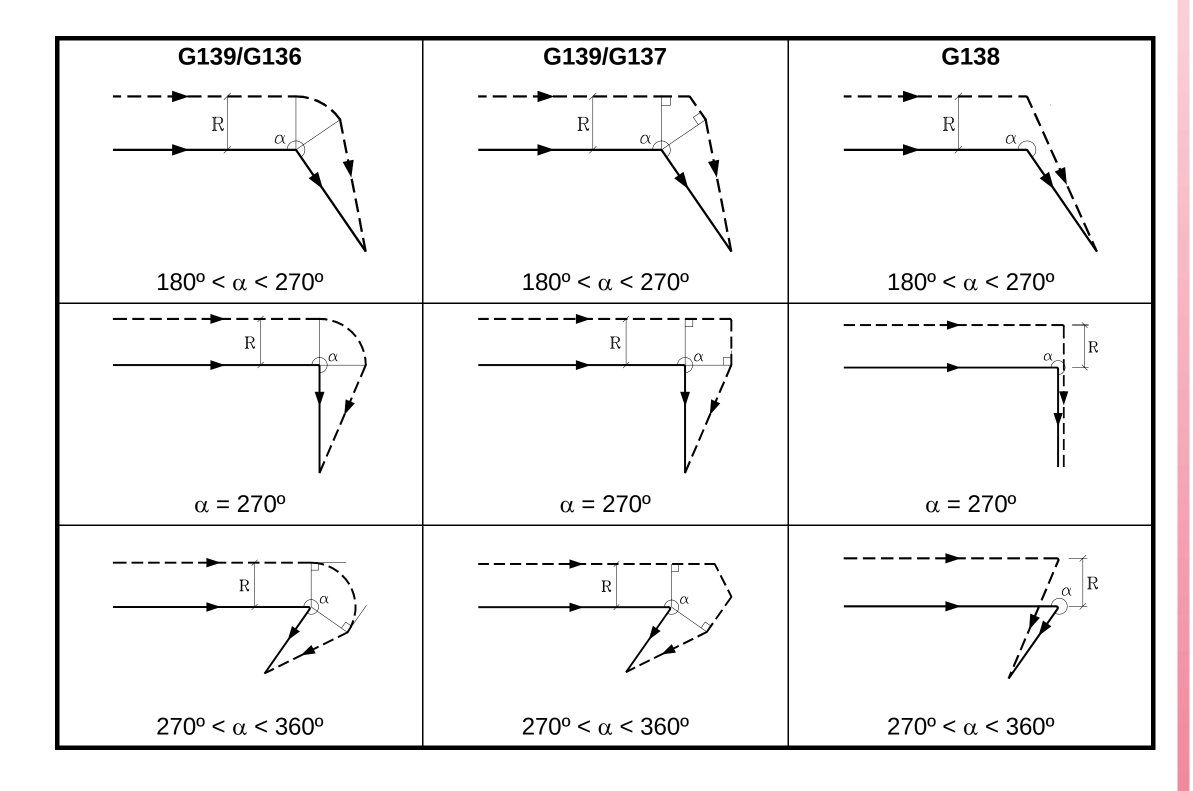


直线 -- 直线路径

当直线和直线的夹角小于或等于 180º 时时，刀具半径补偿撤销时的路径与 功能 G136/G137 和 G138/G139 的选择情况无关。

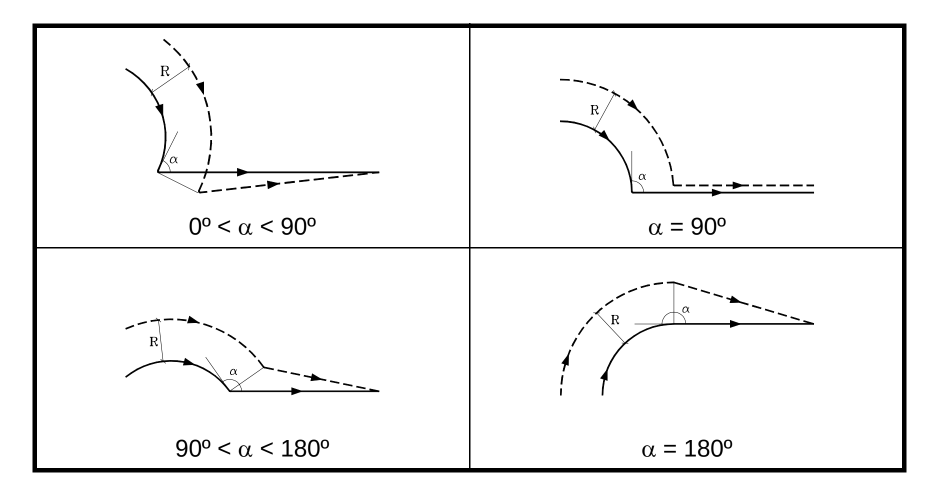


当两路径夹角大于 180º 时，刀具半径补偿撤销时的路径与功能 G136/G137 和 G138/G139 的选择情况有关。

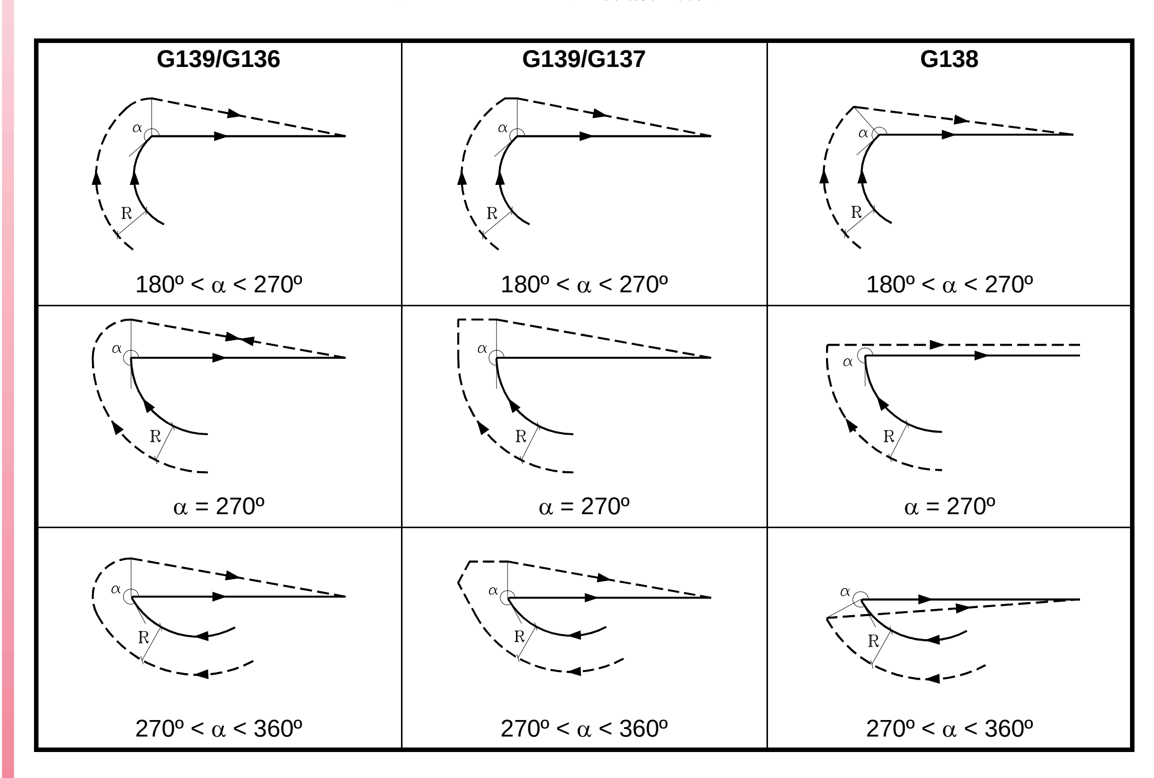


圆弧 -- 直线路径

当圆弧切线和直线的夹角小于或等于 180º 时，刀具半径补偿撤销时的路径与功能 G136/G137 和 G138/G139 的选择情况无关。



当圆弧切线和直线的夹角大于 180º 时，刀具半径补偿撤销时的路径与功能 G136/G137 和 G138/G139 的选择情况有关。



9.2 刀具长度补偿

使用指令 "#TOOL AX" 在轴上应用刀具长度补偿，当缺少时，通过选择平面指定纵向轴为应用刀具长度补偿的轴 。

如果是 G17，刀具长度补偿应用于 Z 轴。

如果是 G18，刀具长度补偿应用于 Y 轴。

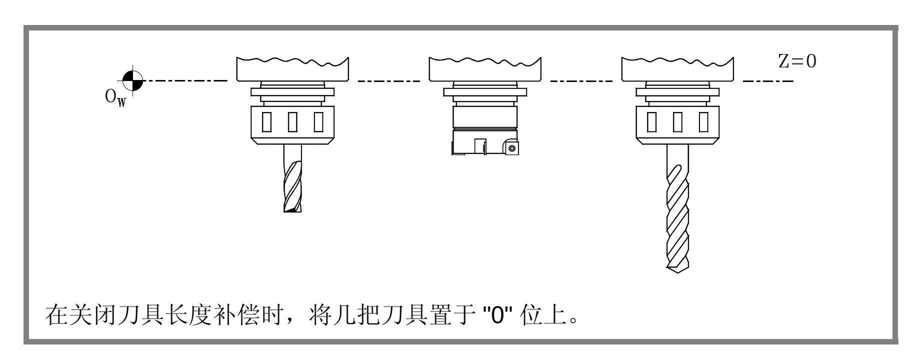
如果是 G19，刀具长度补偿应用于 X 轴。

无论执行 G17, G18 或 G19 中的哪一个，数控系统采用垂直于所选平面的轴作为新的纵向轴。选择后执行 "#TOOL AX"，新选择的纵向轴代替先前选择的。

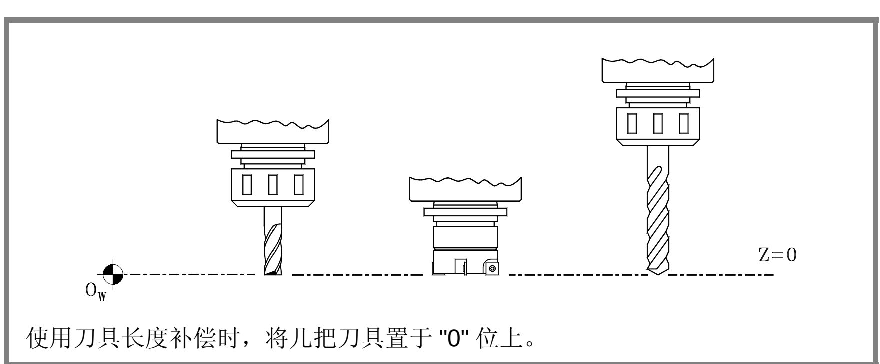
编程

当选择刀具偏置时，激活刀具长度补偿。

- 激活补偿，编写 "D<n>"，其中 <n> 是刀具偏置号，包含用于补偿值的刀具尺寸。
- 编写 "D0"，撤销刀具长度补偿。



在关闭刀具长度补偿时，将几把刀具置于 "0" 位上。



使用刀具长度补偿时，将几把刀具置于 "0" 位上。

一旦执行这些编码中的一个，刀具长度补偿将被激活，或者在纵向轴的下一个运动中被撤销。

铣削固定循环.

10

10.1 概述

一些固定循环用 ISO 代码编辑 （本章介绍），另一些则通过对话模式生成。（“12 循环编辑器”章介绍）

ISO 代码编辑的固定循环用 "G" 功能和相关参数来定义。

G81	钻削循环
G82	变步长往复式钻削循环
G83	恒步长往复式深孔钻削循环
G84	攻丝循环
G85	铰孔循环
G86	镗孔循环
G87	矩形型腔铣削循环
G88	圆柱型腔铣削循环

和固定循环相关的其它功能：

G80	固定循环取消
G98	在固定循环结尾返回到开始平面
G99	在固定循环结尾返回到参考平面

可以在任何平面执行固定循环加工。

铣削和车削固定循环

数控系统提供在同一台机床上使用铣削和车削固定循环的可能，如果机床允许这样。

因为两种类型的固定循环共享相同的 G 功能，可以如下选择要执行的固定循环。缺省时，执行安装的软件固定循环。



在铣削模式的数控系统中 (安装铣削软件).

缺省时，执行铣削固定循环。用以下指令执行车削固定循环：

#LATHECY ON - 激活车削固定循环

#LATHECY OFF - 取消车削固定循环

```
G81 ···              钻孔循环。
#LATHECY ON          激活车削固定循环。
G81 ···
G87 ···
#LATHECY OFF         取消车削固定循环。
```

在车削模式的数控系统中 (安装车削软件).

缺省时，执行车削循环。用以下指令执行铣削固定循环 :

#MILLCY ON - 激活铣削固定循环

#MILLCY OFF - 取消铣削固定循环

```
G81 ···              车削固定循环。
#MILLCY ON           激活铣削固定循环。
G81 ···
G86 ···
#MILLCY OFF          取消铣削固定循环。
```




10.1.1 固定循环定义

固定循环可以定义在程序的任何位置，在主程序或子程序内都可以。

用相应的 "G" 功能和相关的参数来定义。

执行一个固定循环不改变先前 "G" 功能的历史记载，保持主轴旋转方向。如果主轴是停止的，主轴开始顺时针旋转 (M03)。

10.1.2 固定循环的影响范围

固定循环是模态的。一旦通过编程或 MDI 定义，保持有效，直到编写功能取消，或是直到取消它的条件发生。

如果在固定循环的影响范围之内，当激活固定循环时，执行一个运动程序段，开始执行编写的运动，然后执行相应固定循环的加工操作。

实例

```
T1 D1 M6
G0 G90 Z25 S1000 M3 M8 M41 F200
G5 X15 Y15          ( 运动到 X15 Y15)
G99 G81 Z2 I-20     ( 定义并执行钻孔循环 )
X85                 ( 运动，在 X85 Y15 钻新孔 )
Y85                 ( 运动，在 X85 Y85 钻新孔 )
G80                 ( 取消固定循环 )
X15                 ( 运动到 X15 Y85) 不钻孔 )
M30
```

10.1.3 固定循环取消

当在如下情况时取消固定循环 :

- 执行功能 G80 ;
- 定义新的固定循环;
- 用 G20 或 #TOOL AX 选择另一个纵向轴;
- 回零;
- 选择新的工作平面;
- 执行 M02 或 M30 以及在急停或复位后。



10.1.4 工作平面

在固定循环加工中，沿纵向轴有两个坐标很重要，需要说明：

- 初始平面 (Zi)

  当定义固定循环时，刀具坐标 （定位）。

- 参考平面 (Z)

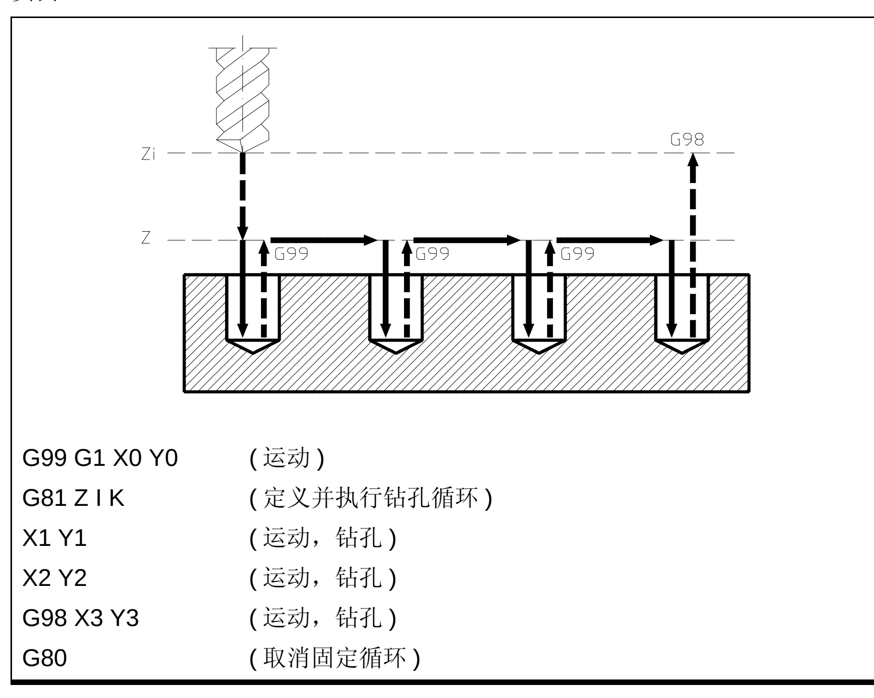
  当定义固定循环时编写它，接近工件的坐标。

功能 G98 和 G99 说明加工结束后刀具返回的位置 。

G98	返回初始平面 (Zi)。
G99	返回参考平面 (Z)。

两个功能都是模态的，缺省时，采用功能 G98 。

实例 :



G99 G1 X0 Y0	(运动)
G81 Z I K	(定义并执行钻孔循环)
X1 Y1	(运动，钻孔)
X2 Y2	(运动，钻孔)
G98 X3 Y3	(运动，钻孔)
G80	(取消固定循环)

10.1.5 编程命令

在定义固定循环之前，必须定义准备功能 (G), 工艺功能 (F, S) 和辅助功能 (M, H) 。

功能 G98, G99 和位置在运动到加工点之前也必须编写。

实例 :

```
T1 D1 M6
    选择刀具 1 和偏置 1。
G0 G90 X0 Y0 Z25
    快速移动刀具到 X0 Y0 Z25。
N10 G99 G1 X60 I30 F1000 S2000 M4。
    在 G1 方式运动到加工点 X60 Y0。
    初始平面 Z25.
    加工结束后退回参考平面 (G99)。
N11 G81 Z2 I-20
    在 X60 Y0 钻孔。
    退回到 Z2，参考平面 (G98 有效 )。
    保持在固定循环前的条件 (G1 F1000 S2000 M4)。
G98 G2 X160 I50
    圆弧插补到 (G2) 到点 X160 Y0 Z25 ;
    在该点钻孔。
    退回到初始平面 (Z25)。
M30
```

程序段 N10 (运动) 和 N11 (固定循环定义) 也可以作为单独程序段定义，在结尾处定义的单独的程序段。

N10 G99 G1 X60 I30 F1000 S2000 M4

N11 G81 Z2 I-20

N10 G99 G1 X60 I30 F1000 S2000 M4 G81 Z2 I-20

在另一个有效的固定循环的影响范围内，定义一个新的固定循环时，用以下方法。

N10	G81 Z2 I-20	N10	G81 Z2 I-20
	X160 I50 F3000		X160 I50 F3000
N20	G80		
N30	G1 X200 Y200	N30	G1 X200 Y200 G83 Z2 I-2 J5
N31	G83 Z2 I-2 J5		
	X220		X220
	M30		M30

在实例的左边，取消有效的固定循环必须编写程序段 N20 。否则，程序段 N30 执行程序段 N10 定义的有效的固定循环。

在实例的右边，不需要编写程序段 N20。当在 N30 定义一个新的固定循环时，取消在程序段 N10 定义的有效固定循环。当执行程序段 N30，先移动轴到 X200 Y200，然后执行固定循环 G83 。



10.1.6 在其它平面编程

下面的例子说明如何在 X 和 Y 两个方向钻孔。

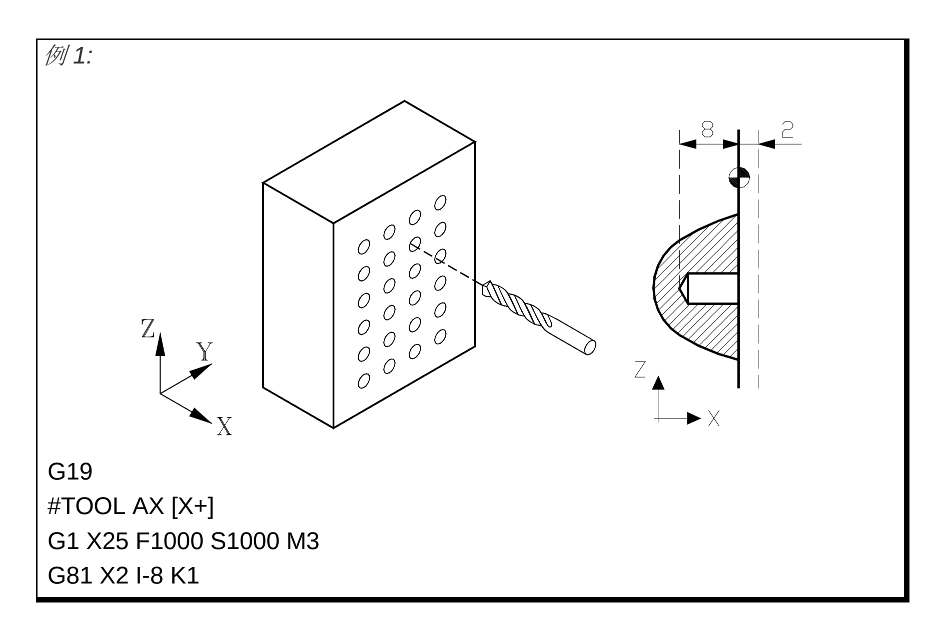
功能 G81 定义钻孔循环 。用参数定义。

X/Y/Z	沿纵向轴的参考坐标
I	钻孔深度
K	孔底暂停

在以下的例子中，工件表面作为 0 坐标，孔深 8 mm，参考坐标在工件表面上方 2 mm。

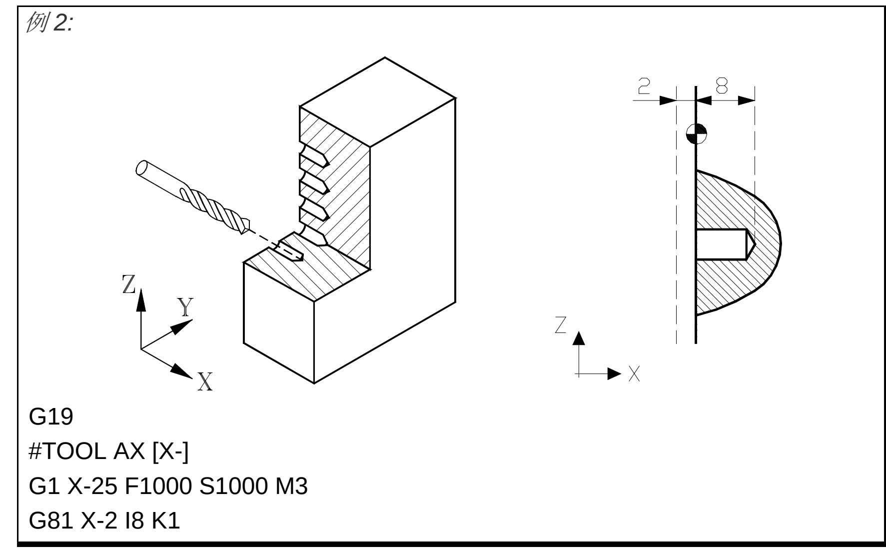
对于每种类型的机床和加工操作，刀具的纵向轴必须用指令 #TOO AX 选择，以使数控系统知道加工方向。

例 1:



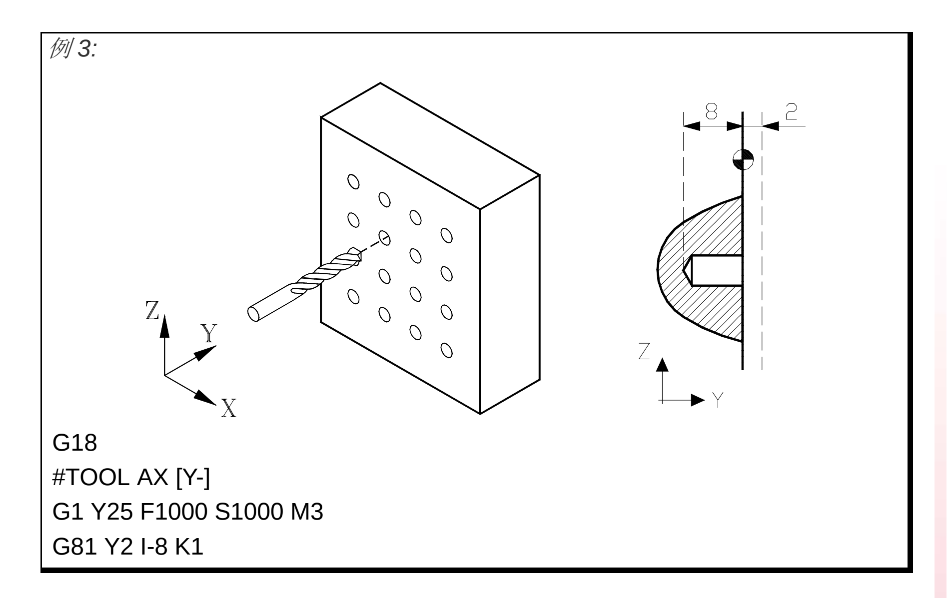

```
G19
#TOOL AX [X+]
G1 X25 F1000 S1000 M3
G81 X2 I-8 K1
```

例 2:



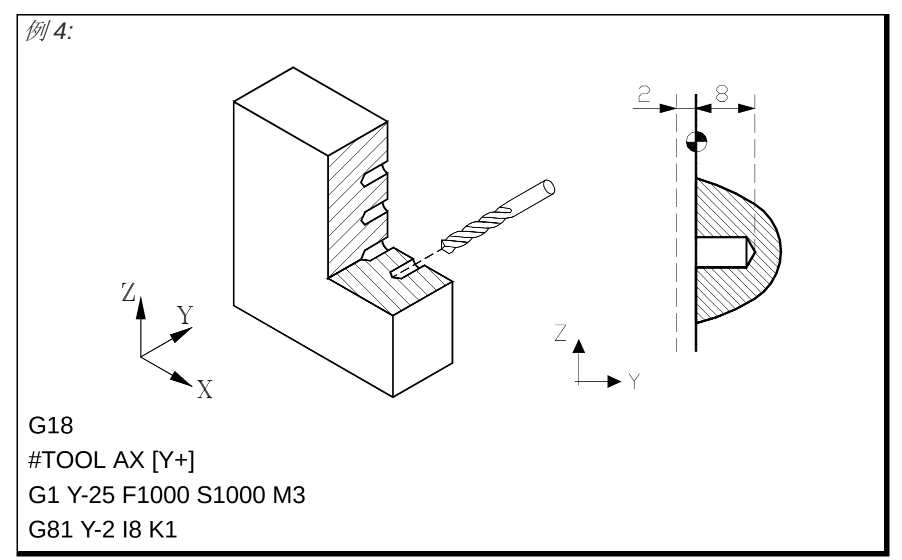

```
G19
#TOOL AX [X-]
G1 X-25 F1000 S1000 M3
G81 X-2 I8 K1
```

例 3:




```
G18
#TOOL AX [Y-]
G1 Y25 F1000 S1000 M3
G81 Y2 I-8 K1
```

例 4:




```
G18
#TOOL AX [Y+]
G1 Y-25 F1000 S1000 M3
G81 Y-2 I8 K1
```

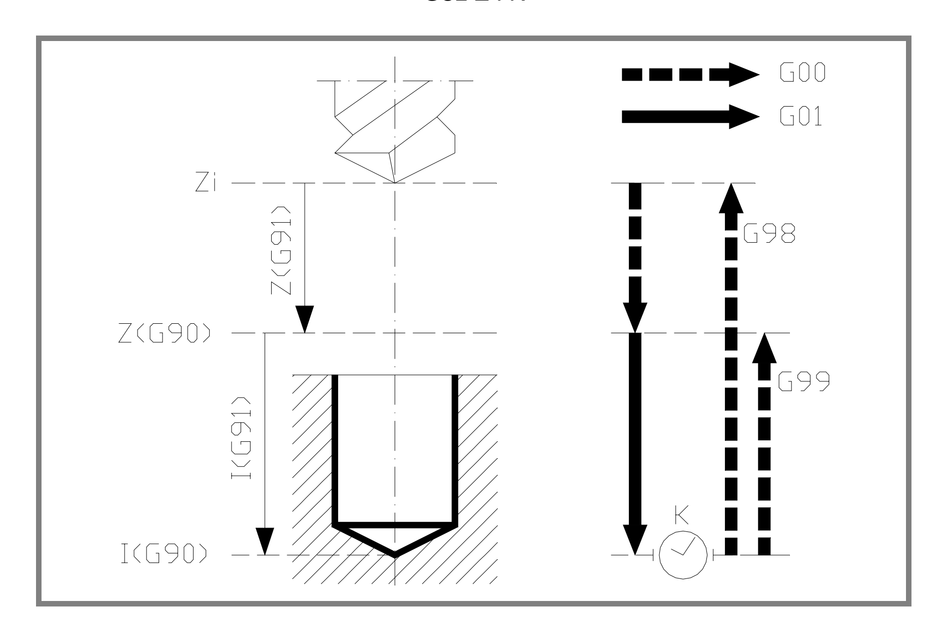
如果工作在 U V 平面，刀具安装在纵向 X2 轴方向。编程方法如下 :

```
#SET AX [U,V,X2]
#TOOL AX [X2+]
G1 X2=25 F1000 S1000
G81 X2=2 I-8 K1
```

10.2 G81. 钻削固定循环

在笛卡尔坐标下的编程格式：

G81 Z I K



参数定义：

Z 参考平面

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi)。

I 钻孔深度。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于参考平面 (Z)。

K 暂停时间，单位为秒，在钻孔和退回运动之间。

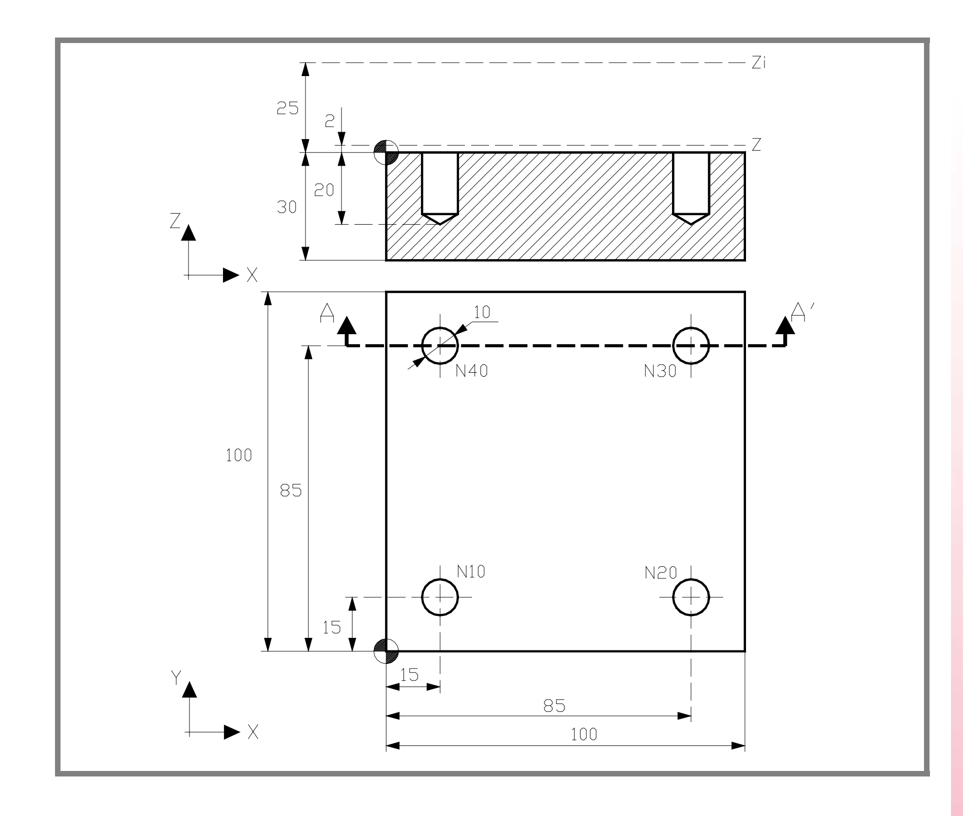
如果没有编写，采用 K0。

基本操作：

1. 如果主轴已经运转，保持运转方向。如果主轴是停止的，主轴开始顺时针旋转 (M03)。
2. 纵向轴 (G0) 从初始平面 (Zi) 快速运动到参考平面 (Z)。
3. 钻孔。纵向轴以工进速度运动到 "I" 编写的孔底位置。
4. 暂停，单位为秒 （如果编写）。
5. 如果功能 G98 有效，快速返回 (G0) 到初始平面 (Zi)；如果功能 G99 有效，快速返回 (G0) 到参考平面 (Z) 。



10.2.1 编程实例



绝对坐标编程 :	
	T1 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 X15 Y15 G81 Z2 I-20
N20	X85
N30	Y85
N40	G98 X15
	M30

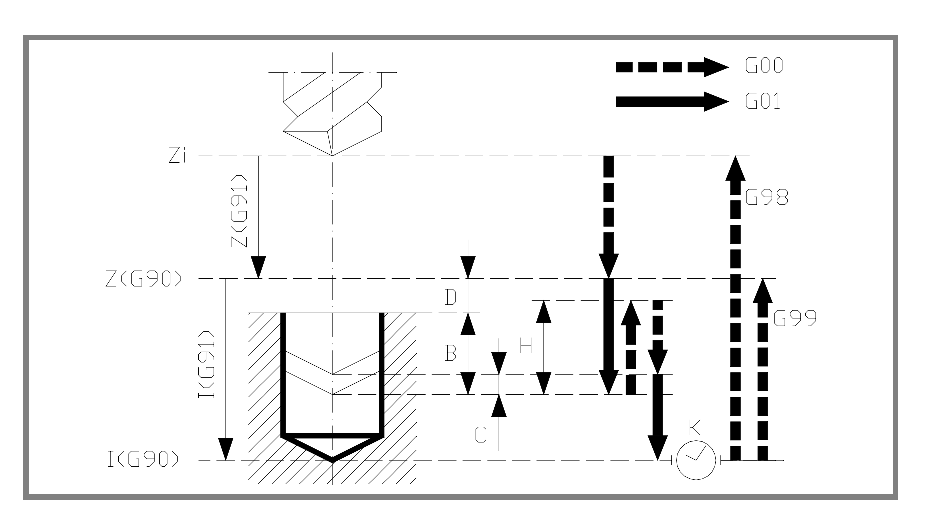
增量坐标编程 :	
	T1 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 G91 X15 Y15 G81 Z-23 I-22
N20	X70
N30	Y70
N40	G98 X-70
	M30



10.3 G82. 变步长往复式钻削循环

在笛卡尔坐标下的编程格式：

G82 Z I D B H C J K R L



参数定义：

Z 参考平面。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi).

I 钻孔深度。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于参考平面 (Zi)。

D 参考平面和工件表面之间的距离。

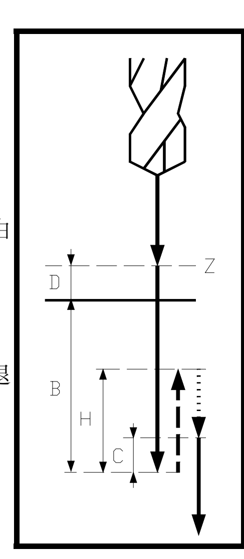
如果没有编写，采用 0 值。

B 单步进给深度。

所有的单步进给深度都是这个值，除了最后一次是由总深度调整。

H 每次单步进给钻孔后，快速 (G0) 返回距离或坐标。

"J"不同于表示距离的0值，"J=0"表示减少的坐标或退回到的绝对坐标。
如果没有编写，返回参考平面。



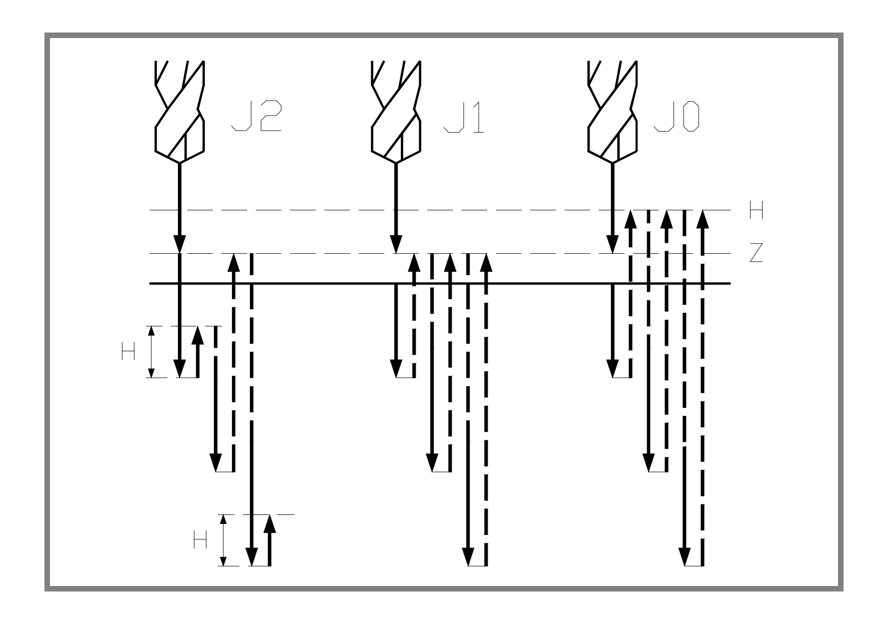
C 接近坐标。

定义从前一次钻削，纵向轴快速运动多大的距离 (G0) 接近零件进行下一次钻削。
如果没有编写，采用 1 mm。
如果编写 "C=0"，系统发出错误报告。



J　　定义在钻入多少以后，刀具以 (G0) 返回参考平面 (Z)。

当 "J" 大于 1 时，每次单步进给后，刀具返回 "H" 指定的距离，每次步长为 "J"，到参考平面 (Z)。
当 "J=1" 时，每次钻入后返回到参考平面 (Z) 。
如果没有编写 "J" 或是 "J=0"，返回 "H" 指定的位置。



K　　暂停时间，单位为秒，在孔的底部。

如果没有定义，采用 0 值。

R　　增大或减小单步进给深度 "B" 的比例因子。

单步进给深度第一步是 "B"，第二步 是 "RB"，第三步是 "R(RB)" 等等。
如果没有编写或 "R=0" ，采用 "R=1"。当 "R=1", 每次单步进给深度值都是 "B" 值。

L　　定义钻入的最小值。用于 "R" 值不是 1 的情况。如果没有编写或编写 0 值，采用值 1 mm。

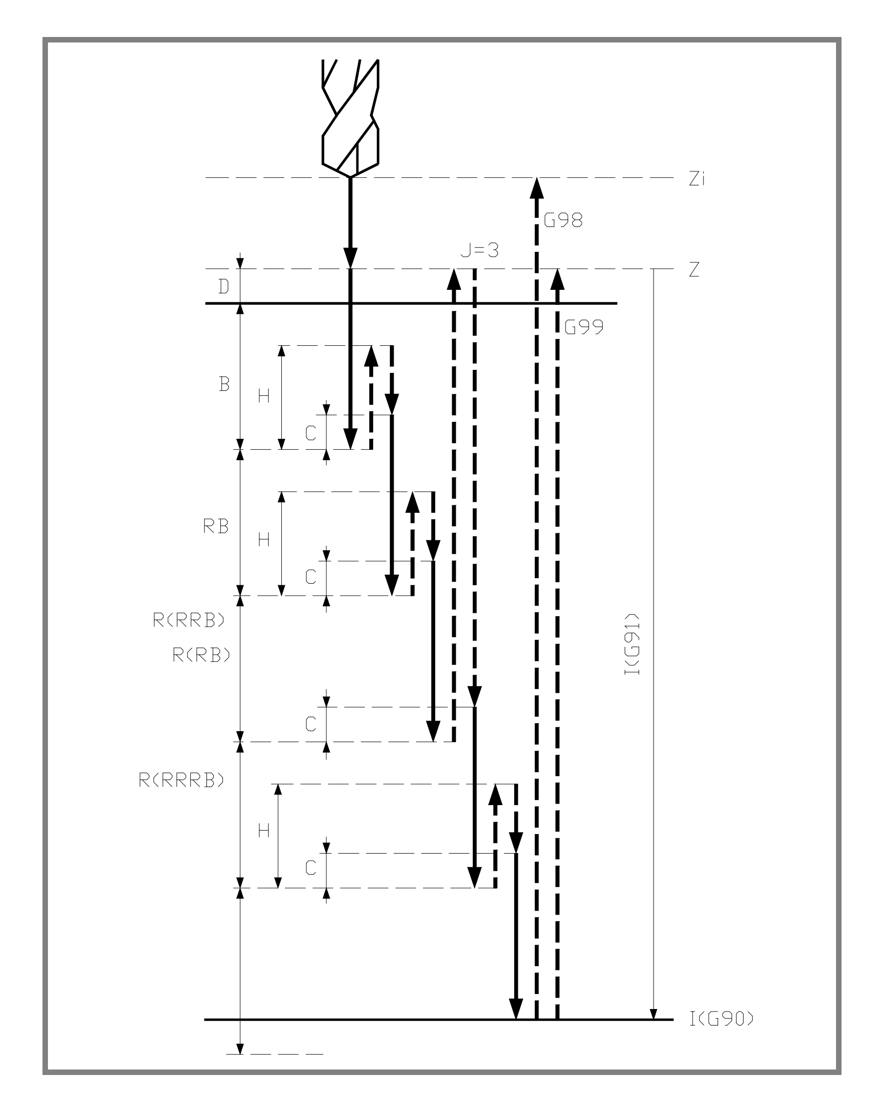
基本操作：

1. 如果主轴先前已经在运动中，将保持原来的转向。如果主轴是停止的，将顺时针启动 (M03)。
2. 纵轴快速 (G0) 从初始平面 (Zi) 运动到参考平面 (Z)。
3. 第一次钻入，按程序编写的工作进给率纵向移动到编程的增量深度 B+D。
4. 钻削循环。下面的步骤将重复进行直到到达编程深度 "I"。
   - 快速退回 (G0).

     当 "J=1" 时 , 每次单步进给钻削后返回参考平面 (Z) 。

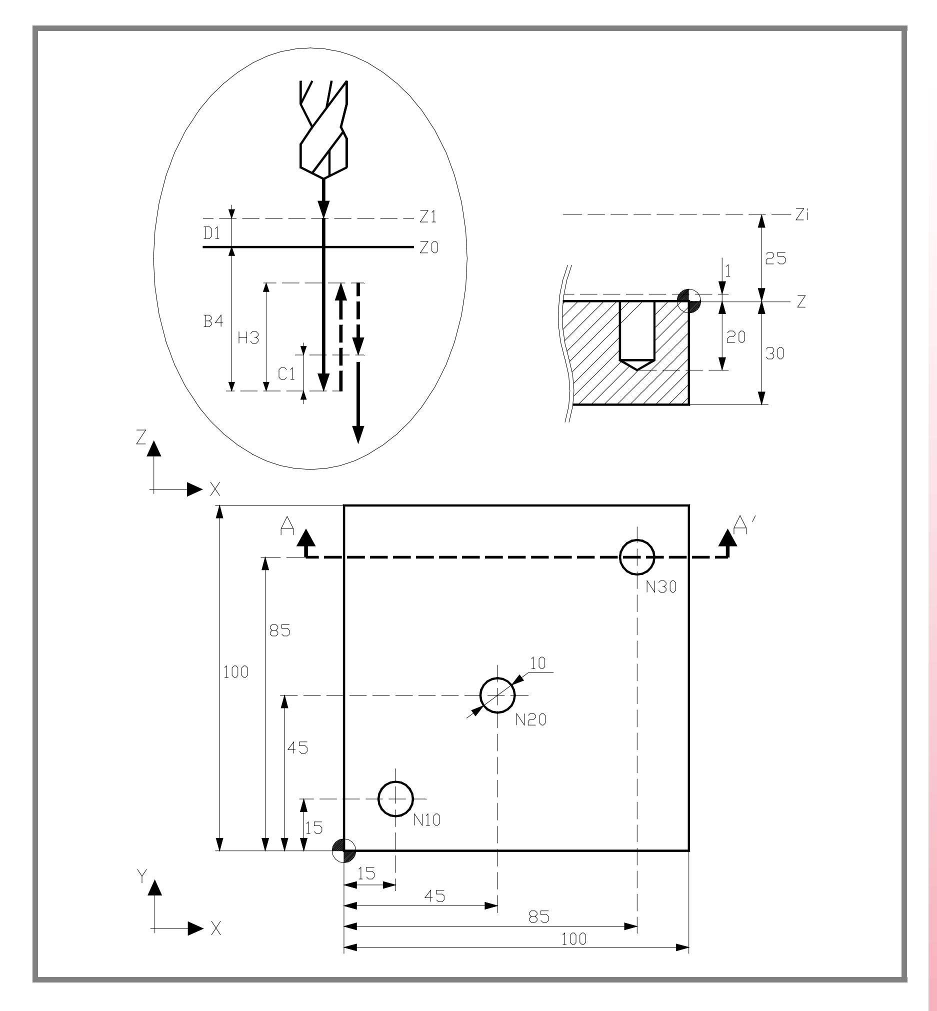
     如果没有编写 "J" 或编写 "J=0" ，每次单步进给钻削后返回 "H" 指定的位置。

     当 "J" 大于 1 时，每次单步进给后，刀具返回 "H" 指定的距离，每次步长为 "J"，到参考平面 (Z)。
   - 快速接近 (G0) 距离 "C" 或距先前钻削深度等于 1 mm。
   - 新的单步进给钻削，纵向轴以工进速度运动，距离由 "B" 和 "R" 给定。



5. 孔底暂停。时间用 "K" 表示，单位秒。
6. 如果功能 G98 有效，快速 (G0) 退回到初始平面 (Zi)；如果功能 G99 有效，快速 (G0) 退回到参考平面 (Z) 。

10.3.1 编程实例



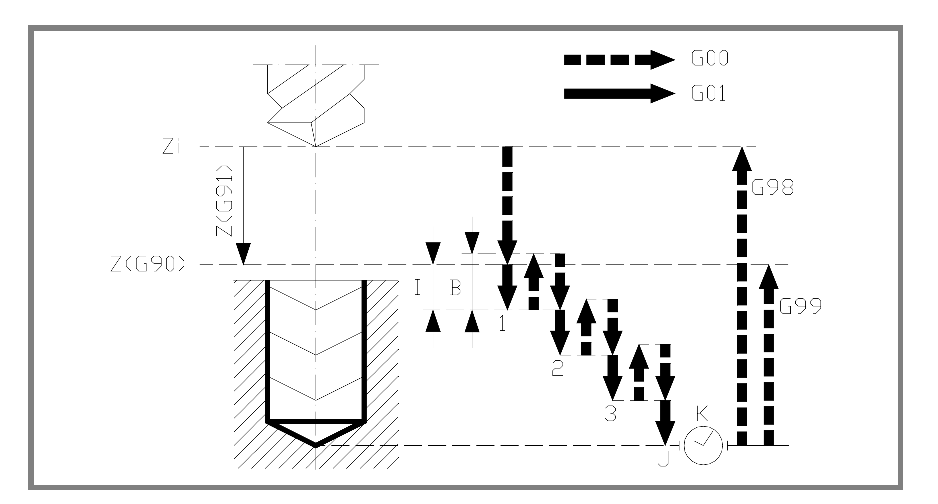

```
绝对坐标编程：
        T2 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 X15 Y15
        G82 Z1 I-20 D1 B4 H3 C1 J3 K1 R0.8 L3
N20     X45 Y45
N30     G98 X85 Y85
        M30
```

```
增量坐标编程：
        T2 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 G91 X15 Y15
        G82 Z-24 I-21 D1 B4 H3 C1 J3 K1 R0.8 L3
N20     X30 Y30
N30     G98 X40 Y40
        M30
```

10.4 G83. 恒步长往复式深孔钻削循环

在笛卡尔坐标下的编程格式 :

G83 Z I J B K



参数定义 :

Z 参考平面。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi)。

I 单步钻孔深度。

符号表示加工方向。正号朝正方向，符号朝负方向。 "I-".

J 定义钻孔操作需要的钻孔的步数。

B 每次单步钻孔后快速退回 (G0) 的距离 。

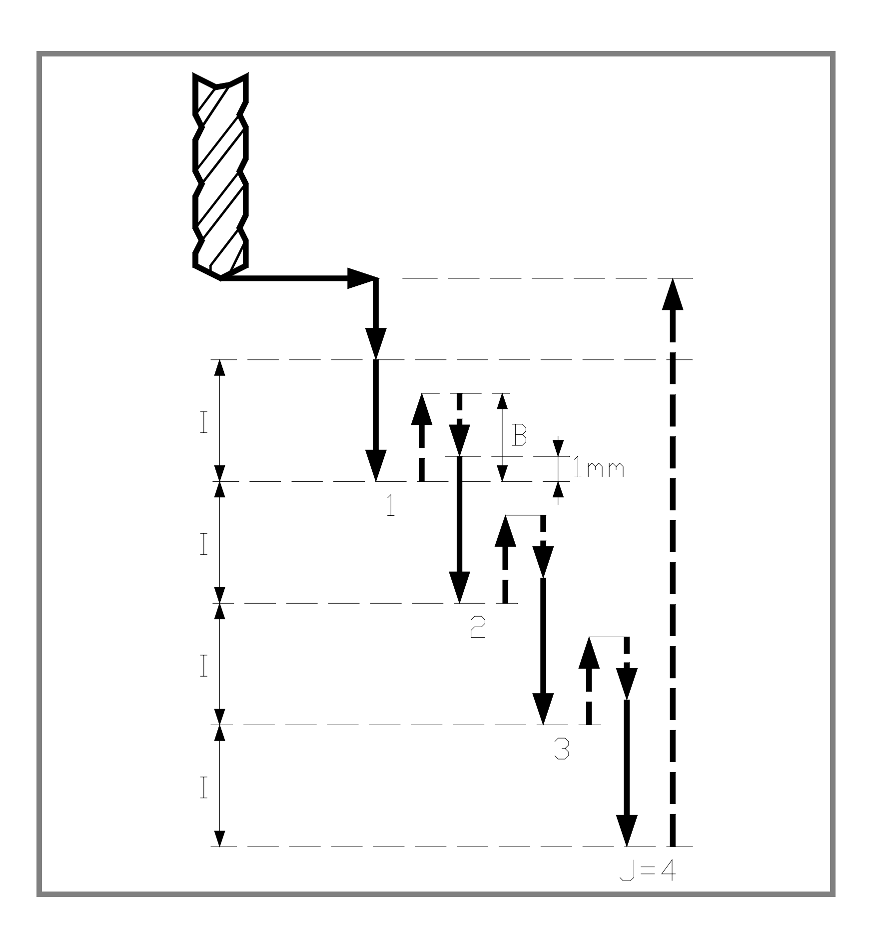
如果没有编写，退回参考平面。

K 在孔底的暂停时间，单位为秒。

如果没有定义，采用 0 值。

基本操作 :

1. 如果主轴先前已经运动，将保持原来的转向。如果主轴是停止的，将顺时针启动 (M03)。
2. 纵轴快速 (G0) 从初始平面 (Zi) 运动到参考平面 (Z)。
3. 钻削循环。下面的步骤将重复进行 "J" 次。
   - 以工进速度单步钻孔，到 "I" 确定的距离。
   - 快速 (G0) 退回 "B" 的距离或到参考平面。
   - 快速 (G0) 接近距前次钻深 1 mm 的地方。

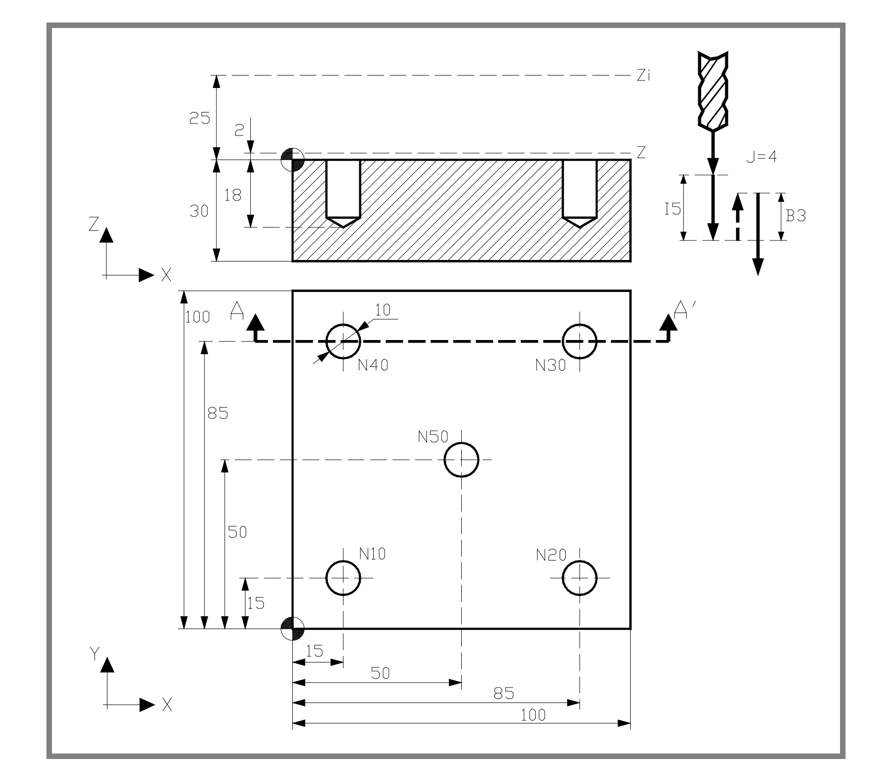


4. 孔底暂停。 "K" 表示时间，单位为秒。

5. 如果功能 G98 有效，快速 (G0) 退回到初始平面 (Zi)；如果功能 G99 有效，快速 (G0) 退回到参考平面 (Z) 。



10.4.1 编程实例



绝对坐标编程 :

```
        T3 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 X15 Y15
        G83 Z2 I-5 J4 B3 K1
N20     X85
N30     Y85
N40     X15
N50     G98 X50 Y50
        M30
```

增量坐标编程 :

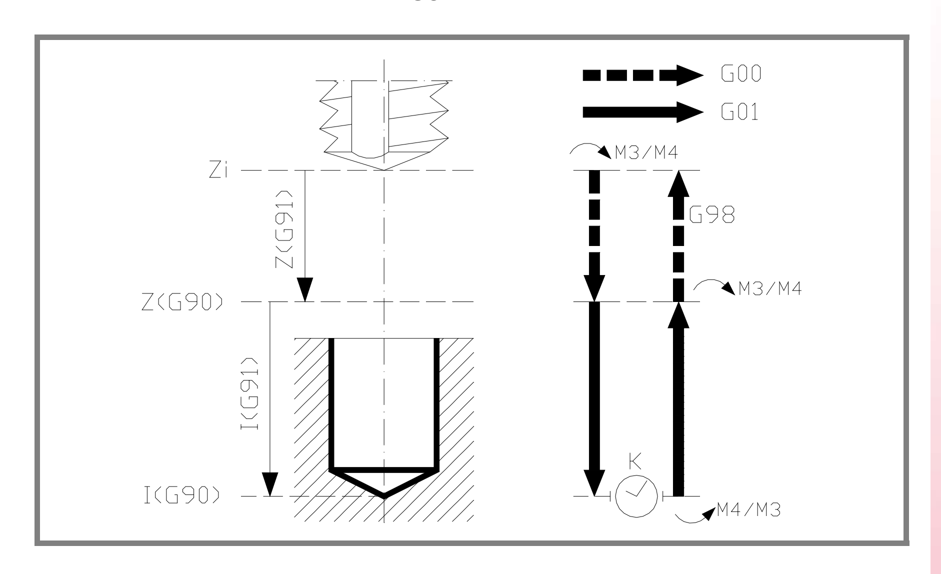
```
        T3 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 G91 X15 Y15
        G83 Z-23 I-5 J4 B3 K1
N20     X70
N30     Y70
N40     X -70
N50     G98 X35 Y-35
        M30
```

10.5 G84. 攻丝固定循环

可以进行攻丝和刚性攻丝。对于刚性攻丝，主轴必须安装编码器和伺服驱动

在笛卡尔坐标下的编程格式 :

G84 Z I K R



参数定义 :

Z　　参考平面。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi)。

I　　攻丝深度。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于参考平面 (Z)。

K　　暂停，单位为秒。在攻丝和退回运动之间。

如果没有编写，采用 0 值。

R　　攻丝类型。

R0: 常规攻丝。
R1: 刚性攻丝。



基本操作 :

1. 如果主轴先前已经运动，将保持原来的转向。如果主轴是停止的，将顺时针启动 (M03)。
2. 纵轴快速 (G0) 从初始平面 (Zi) 运动到参考平面 (Z)。
3. 攻丝。执行 100% 编写的进给率和主轴转速。攻丝不能被中断。
4. 如果 If "K" 不是 0, 主轴停止并暂停。
5. 主轴反方向旋转。

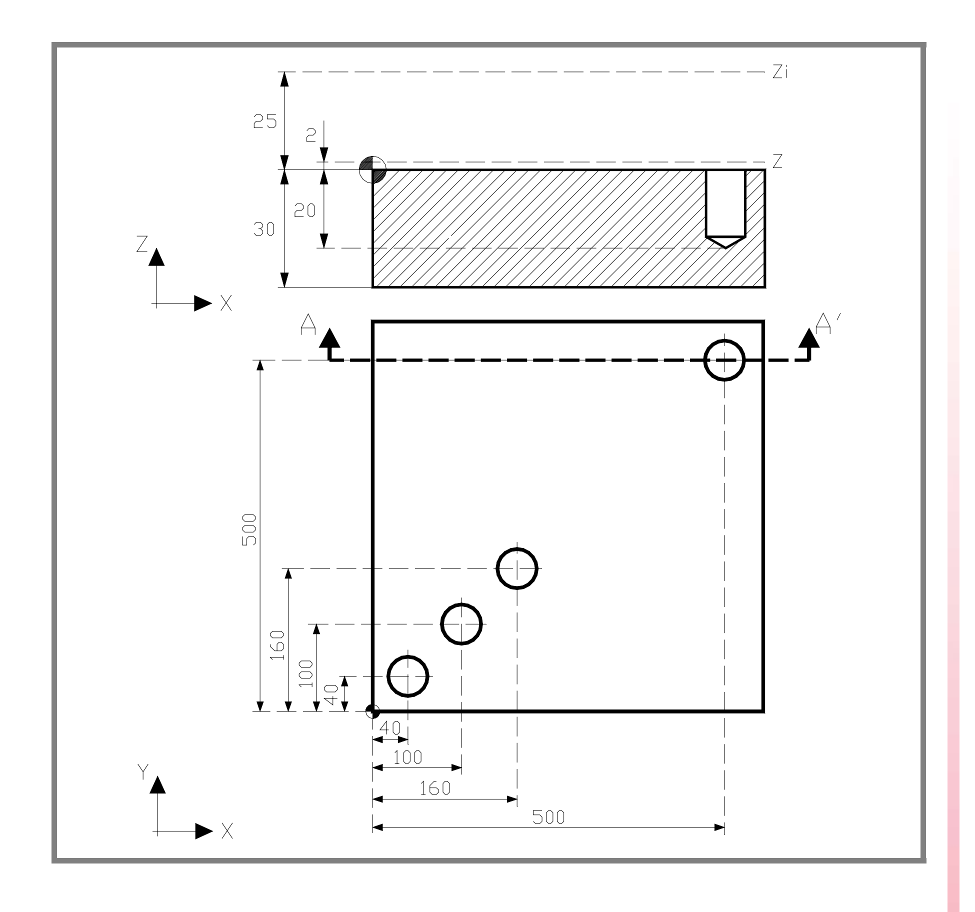
   退回，推出攻丝刀具到参考平面，以 100% 编写的进给率 "F" 和主轴转速 "S" 。刀具退出不能被中断。
6. 根据编写的攻丝类型 .

   R=0 主轴反方向旋转，恢复初始旋转方向。

   R=1 主轴定位 (M19)。
7. 如果 G98 有效，快速返回初始平面 (Zi).



10.5.1 编程实例



绝对坐标编程 :	
	T4 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 X40 Y40 G84 Z2 I-20 K1 R0
N20	X100 Y100
N30	X160 Y160
N40	G98 X500 Y500
	M30

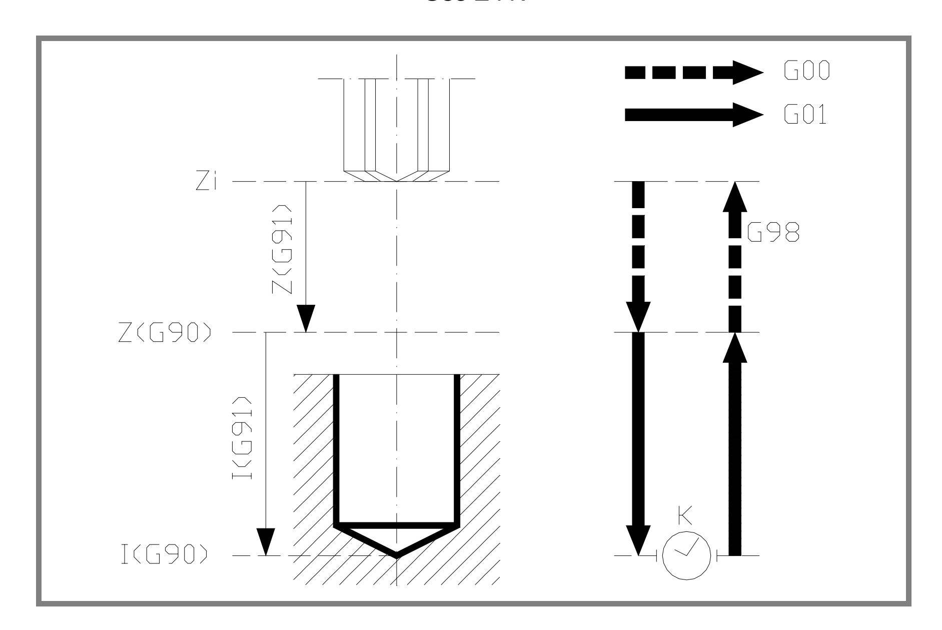
增量坐标编程 :	
	T4 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 G91 X40 Y40 G84 Z-23 I-22 K1 R0
	$FOR P0=1,2,1
	X60 Y60
	$ENDFOR
	G98 X340 Y340
	M30

10.6 G85. 铰孔固定循环



在笛卡尔坐标下编程格式 :

G85 Z I K



参数定义 :

Z 参考平面。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi)。

I 铰孔深度。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于参考平面 (Z)。

K 暂停，单位为秒。在铰孔和退回运动之间。

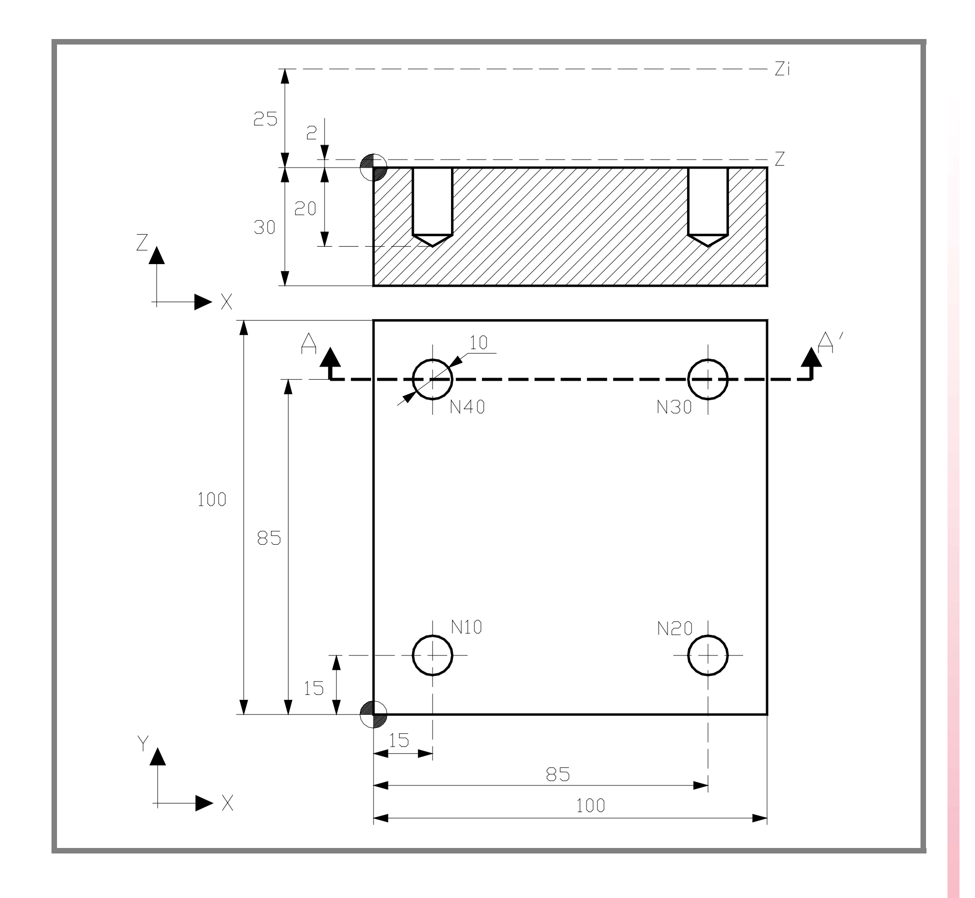
如果没有编写，采用 K0。

基本操作 :

1. 如果主轴先前已经运动，将保持原来的转向。如果主轴是停止的，将顺时针启动 (M03)。
2. 纵轴快速 (G0) 从初始平面 (Zi) 运动到参考平面 (Z)。
3. 铰孔。纵向轴以工进速度运动到 "I" 编写的孔底位置。
4. 暂停，单位为秒 （如果编写）。
5. 退回，以工进速度 (G01) 一直运动到参考平面 (Z)。
6. 如果功能 G98 有效，快速退回到初始平面 (Zi)。



10.6.1 编程实例




```
绝对方式编程:
        T5 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 X15 Y15 G85 Z2 I-20
N20     X85
N30     Y85
N40     G98 X15
        M30
```

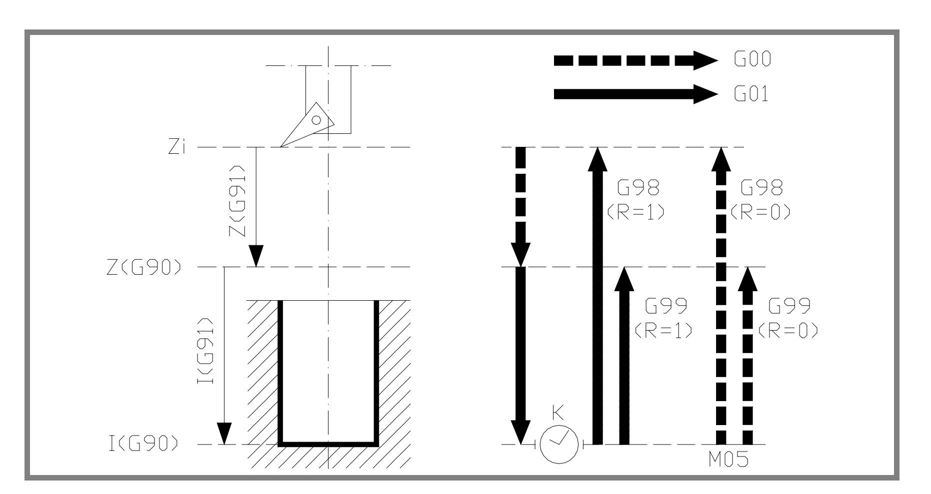
```
增量方式编程:
        T5 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 G91 X15 Y15 G85 Z-23 I-22
N20     X70
N30     Y70
N40     G98 X-70
        M30
```




10.7 G86. 镗孔固定循环

在笛卡尔坐标下的编程格式 :

G86 Z I K R



参数定义 :

Z 参考平面。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi)。

I 镗孔深度。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于参考平面 (Z)。

K 暂停，单位秒。在镗孔和退回运动之间。

如果没有编写，采用 K0。

R 退回类型 : R0 快进 (G0)， R1 工进速度 (G01)。缺省时， R0。

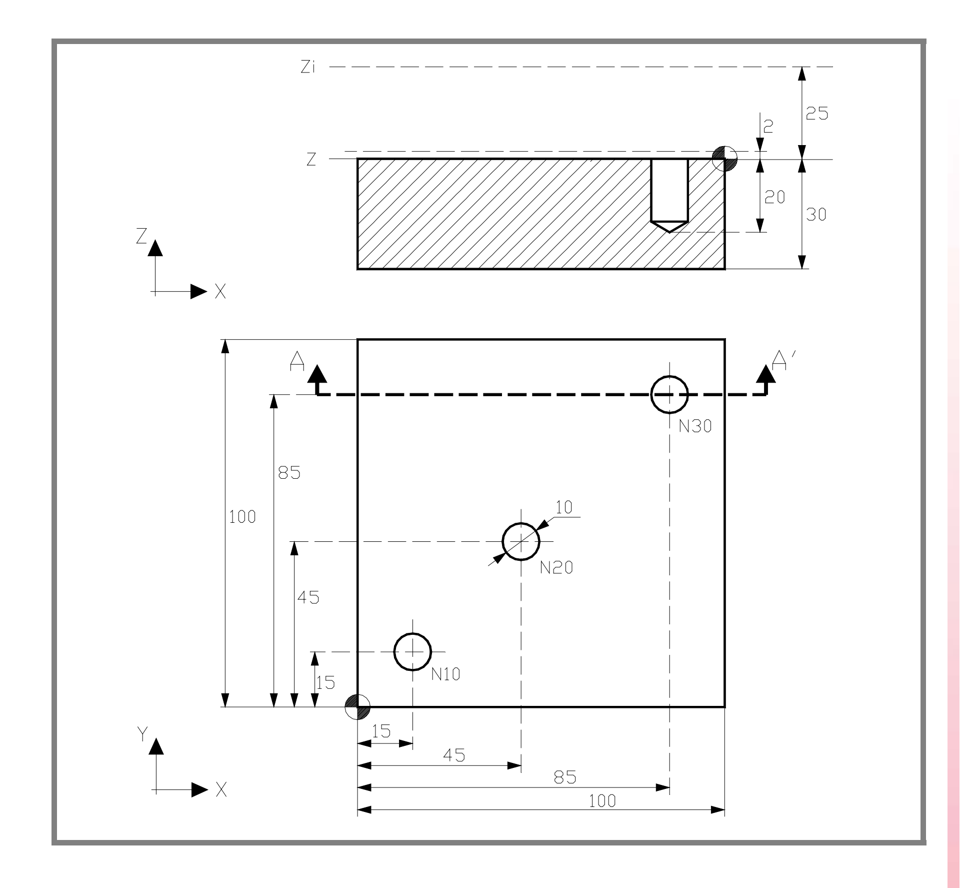
基本操作 :

1. 如果主轴先前已经运动，将保持原来的转向。如果主轴是停止的，将顺时针启动 (M03)。
2. 纵轴快速 (G0) 从初始平面 (Zi) 运动到参考平面 (Z)。
3. 镗孔。纵向轴以工进速度运动到 "I" 编写的孔底位置。
4. 暂停，单位为秒 （如果编写）。
5. 如果编写了 "R=0" ， 主轴停止 (M05)。
6. 如果功能 G98 有效，退回到初始平面 (Zi)；如果功能 G99 有效，退回到参考平面 (Z) 。

   如果 "R=0" 则是快进 (G0) ， 如果 "R=1" 则是工进速度运动 (G01) 。



10.7.1 编程实例



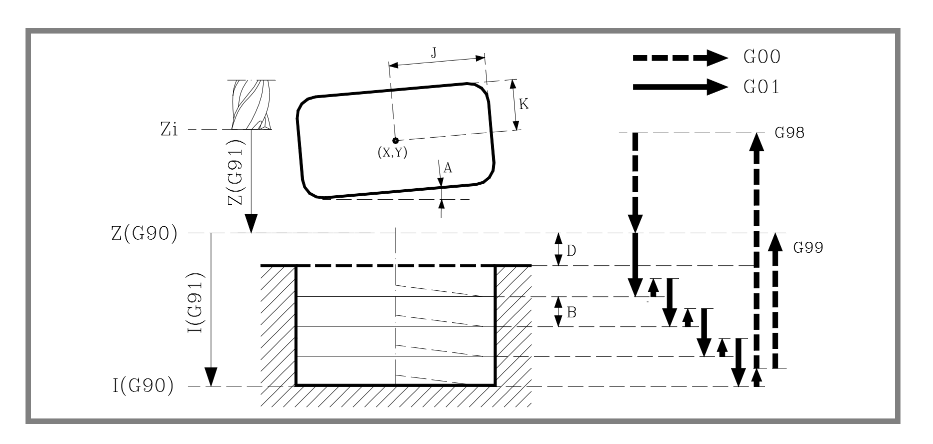

```
绝对方式编程，R=0:
        T6 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 X15 Y15 G86 Z2 I-20 K3 R0
N20     X45 Y45
N30     G98 X85 Y85
        M30
```

```
增量方式编程， R=1:
        T6 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 G91 X15 Y15 G86 Z-23 I-22 K3 R1
N20     X30 Y30
N30     G98 X40 Y40
        M30
```

10.8 G87. 矩形腔加工固定循环

在笛卡尔坐标下的编程格式 :

G87 Z I D A J K M Q B C L H V



参数定义 :

Z 参考平面。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi)。

I 定义型腔加工深度。

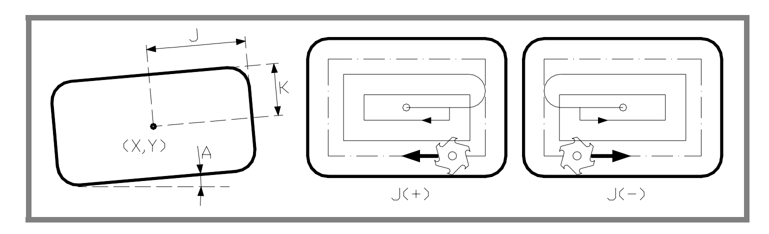
在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于参考平面 (Z)。

D 参考平面和工件表面的距离，没有编写，采用 0 值。

A 型腔和横坐标轴之间的夹角，单位度。没有编写，采用 0 值。

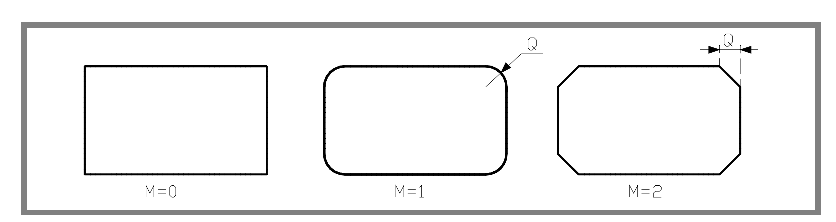
J 型腔长度值的一半。

符号表示型腔加工的方向 :
(J+) 顺时针， (J-) 逆时针。



K 型腔宽度值的一半。

M 拐角类型。 (0) 方角 , (1) 圆角 , (2) 倒角。如果没有编写，采用 0 值。



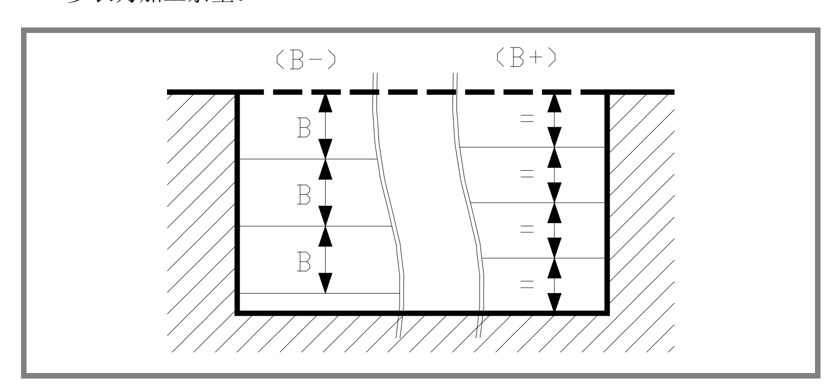


Q 圆弧半径或倒角尺寸。

B 定义沿纵向轴的切削深度。

如果用正号编写 (B+)，整个循环将用相同的步长走刀，该步长等于或小于编程值。

如果用负号编写 (B-)，整个型腔将用给定的步长走刀，最后一次例外，它的步长为加工余量。

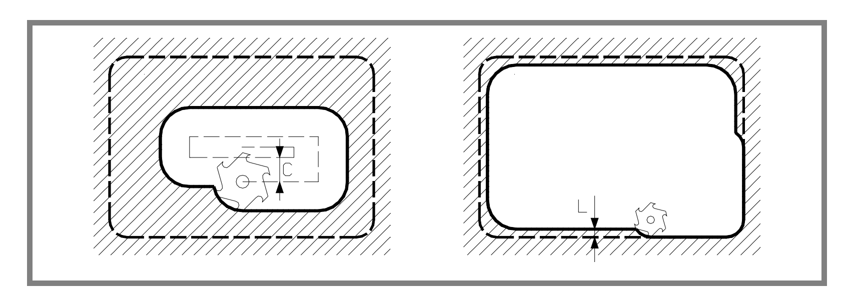


C 铣削路径或宽度。

如果没有编写或编写 0 值，值为所选刀具直径的 3/4 。

如果值同参数 "J" 或 "K" (型腔长度 / 宽度的一半)，仅进行精加工的路径。

如果编写的值大于刀具直径，数控系统发出相关错误报告。



L 定义精加工量。

如果没有编写或编程值为 0 值，不执行精加工。

H 精加工路线进给率。如果没有编写或编写 0 值，执行粗加工进给率。

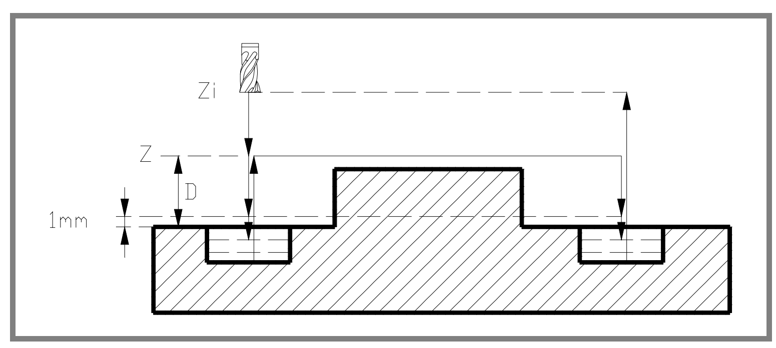
V 定义刀具切入进给率。如果没有编写或编写 0 值，执行平面加工进给率的 50% 。

基本操作 :

1. 如果主轴先前已经运动，将保持原来的转向。如果主轴是停止的，将顺时针启动 (M03)。
2. 纵轴快速 (G0) 从初始平面 (Zi) 运动到参考平面 (Z)。
3. 纵轴快速 (G0) 运动距工件表面 1 mm 的地方。



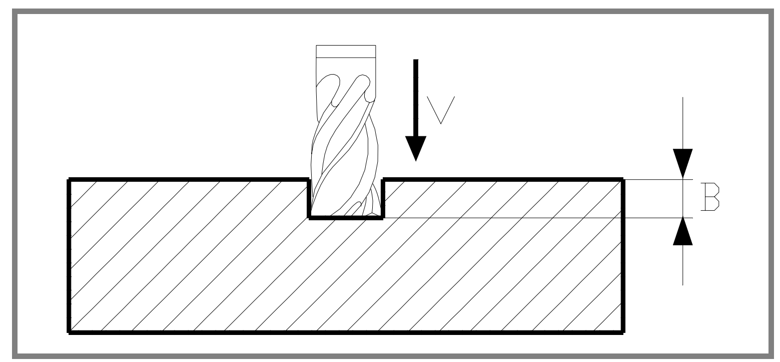
在下图所示的形状中，当安全坐标远离工件表面时，允许快速接近加工表面。 .



4. 第一次切入操作。纵向轴以 "V" 指定的进给率切入工件至 "B" 确定的距离。

5. 以工进速度用 "C" 定义的步长铣削型腔表面，直到离精加工型腔壁距离为 "L"。

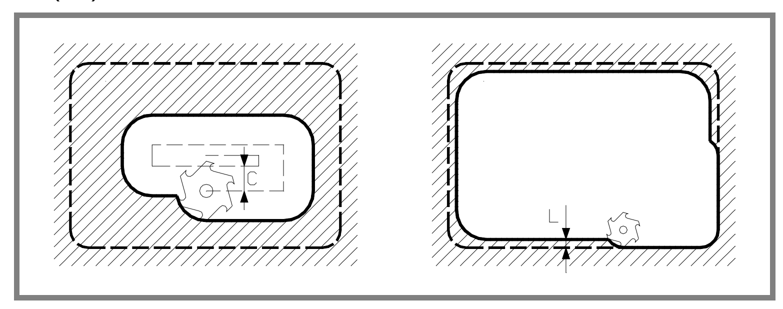
按照参数 "J" 确定的方向执行。



6. 以 "H" 定义的精加工进给率，铣削精加工走刀 "L" 。

为了得到好的工件表面质量，当加工型腔壁时，精加工路线执行切向进入和退出。

7. 快速 (G0) 退回型腔中心，纵向轴距加工表面 1 mm 处。 .



8. 进行下面的加工，直到到达型腔的总深度。

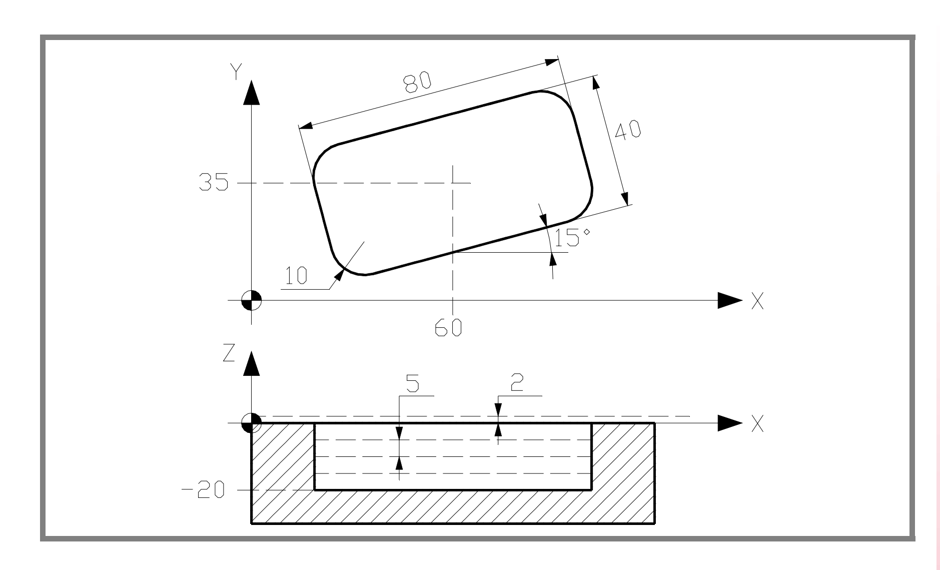
- 切入工件，纵向轴以 "F" 指定的进给率切入工件至 "B" 确定的距离。
- 同步骤 5, 6 和 7 一样，铣削新的表面。

9. 如果功能 G98 有效，退回到初始平面 (Zi)；如果功能 G99 有效，退回到参考平面 (Z) 。



10.8.1 编程实例

加工一个 80x40 的型腔，中心在 (X60, Y35)，旋转角度 15º。型腔上表面是为 Z0，型腔加工到 Z-20。参考平面设在 Z2。



G90 G0 X60 Y35
G87 Z2 I-20 D2 A15 J40 K20 ·····

型腔的拐角是半径为 10 mm 的圆角。

G87 Z2 I-20 D2 A15 J40 K20 **M1 Q10** ·····

切深是 5 mm，以进给率 50 mm/min 执行。

G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 **B5** ····· **V50**

粗加工路径铣削宽度是 5 mm，进给率 800 mm/min。在执行循环前必须选择铣削进给率，所以定义在前一程序段。

G90 G0 X60 Y35 **F800**
G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 B5 **C5** ····· V50

保留精加工余量 1 mm，精加工将以 300 mm/min 的进给率执行。

G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 B5 C5 **L1 H300** V50

现在说明如何执行型腔加工，加工在 (X200 Y135) 和 (X350 Y235) 多个点重复。

绝对方式编程 :

```
      T7 D1 M6
      G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F800
N10   G99 X60 Y35
      G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 B5 C5 L1 H300 V50
N20   X200 Y135
N30   G98 X350 Y235
      M30
```



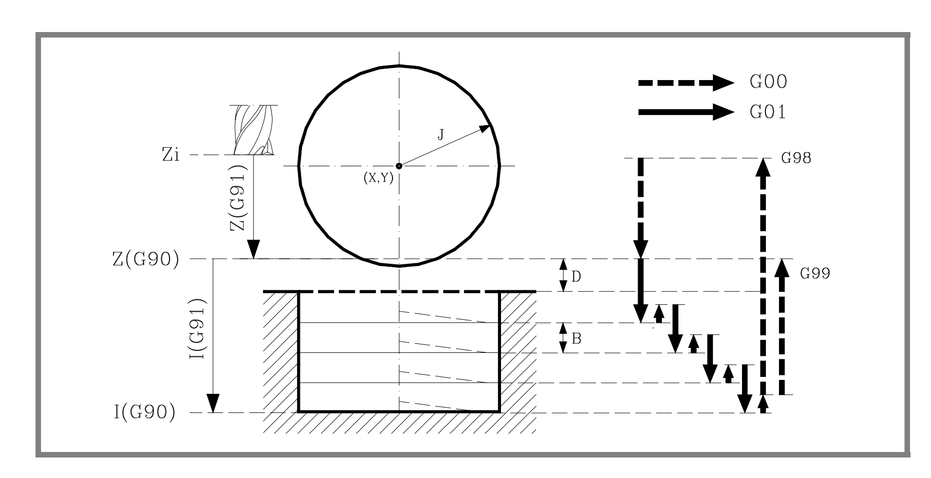

```
增量方式编程 :
      T7 D1 M6
      G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F800
N10   G99 G91 X60 Y35
      G87 Z-23 I-45 D2 A15 J40 K20 M1 Q10 B5 C5 L1 H300 V50
N20   X140 Y100
N30   G98 X150 Y100
      M30
```



10.9 G88. 圆柱型腔加工固定循环

在笛卡尔坐标下的编程格式 :

G88 Z I D J B C L H V



参数定义 :

Z 参考平面。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于初始平面 (Zi)。
如果没有编写，采用刀具当前位置作为参考平面 (Z=Zi)。

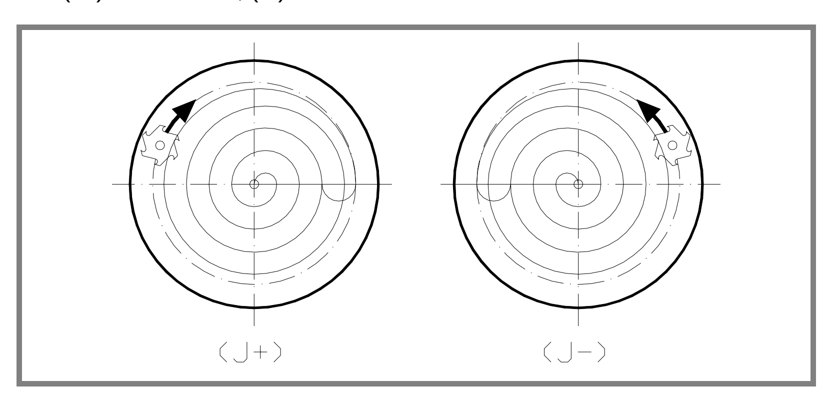
I 型腔加工深度。

在 G90 模式下，坐标相对于工件零点。
在 G91 模式下，坐标相对于参考平面 (Z)。

D 参考平面和工件表面的距离，没有编写，采用 0 值。

J 型腔半径。

符号决定型腔加工方向：
(J+) 顺时针方向 , (J-) 逆时针方向。

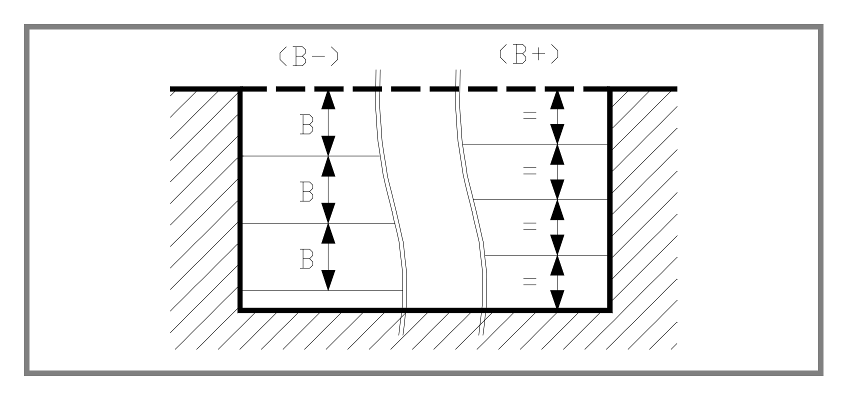




B 定义沿纵向轴的切削深度。

如果用正号编写 (B+)，整个循环将用相同步长走刀，该步长等于或小于编程值。

如果用负号编写 (B-)，整个型腔将用给定步长走刀，最后一次走刀例外，它的步长为加工余量。

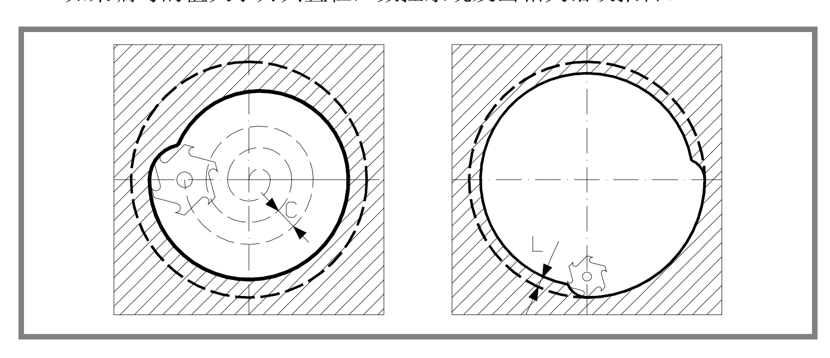


C 铣削路径或宽度。

如果没有编写或编写 0 值，值为所选刀具直径的 3/4 。

如果值同参数 "J" 或 "K" (型腔长度 / 宽度的一半)，仅进行精加工的路径。

如果编写的值大于刀具直径，数控系统发出相关错误报告。



L 定义精加工量。

如果没有编写或编写 0 值，不执行精加工。

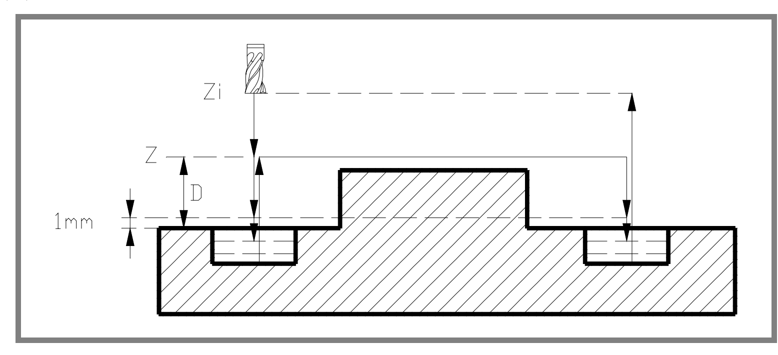
H 精加工进给率。如果没有编写或编程值为 0 ，执行粗加工进给率。

V 刀具切入进给率。如果没有编写或编程值为 0 ，执行在平面的进给率的 50% 。

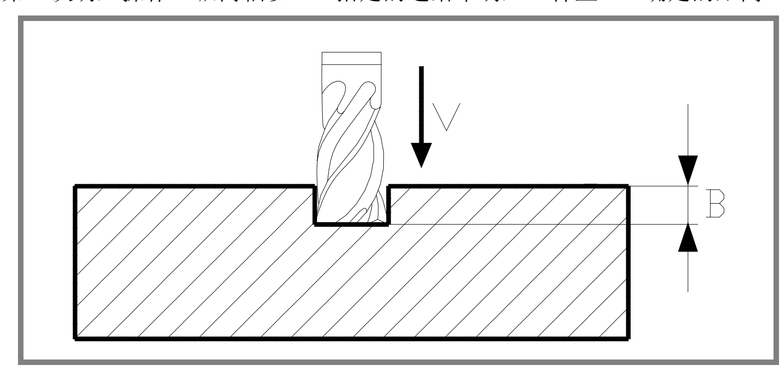
基本操作：

1. 如果主轴先前已经运动，将保持原来的转向。如果主轴是停止的，将顺时针启动 (M03)。
2. 纵轴快速 (G0) 从初始平面 (Zi) 运动到参考平面 (Z)。
3. 纵轴快速 (G0) 运动至距工件表面 1 mm 处。

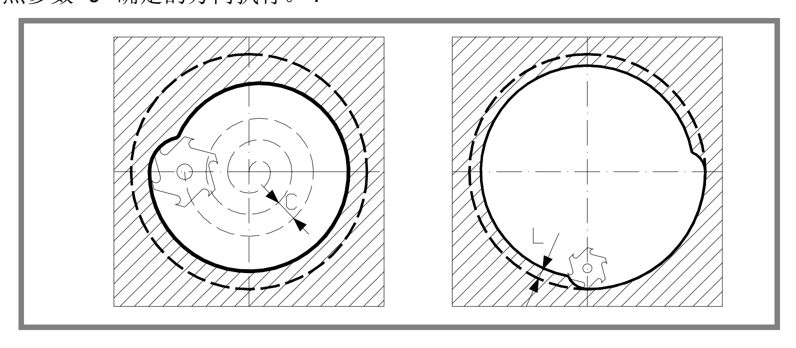
   在下图所示的情况中，当安全坐标 (Z) 远离工件表面时，允许快速运动接近加工表面。 .



4. 第一次切入操作。纵向轴以 "V" 指定的进给率切入工件至 "B" 确定的距离。

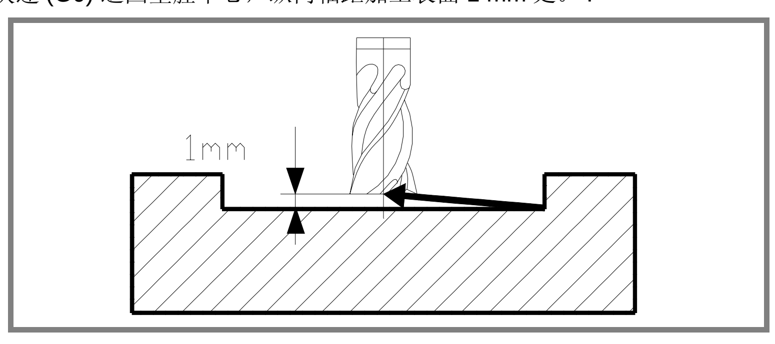


5. 以工进速度用 "C" 定义的步长铣削型腔表面，直到离精加工型腔壁距离为 "L"。按照参数 "J" 确定的方向执行。 .



6. 以 "H" 定义的精加工进给率，铣削精加工走刀 "L" 。

   为了得到好的工件表面质量，当加工型腔壁时，精加工路线执行切向进入和退出。
7. 快速 (G0) 退回型腔中心，纵向轴距加工表面 1 mm 处。 .





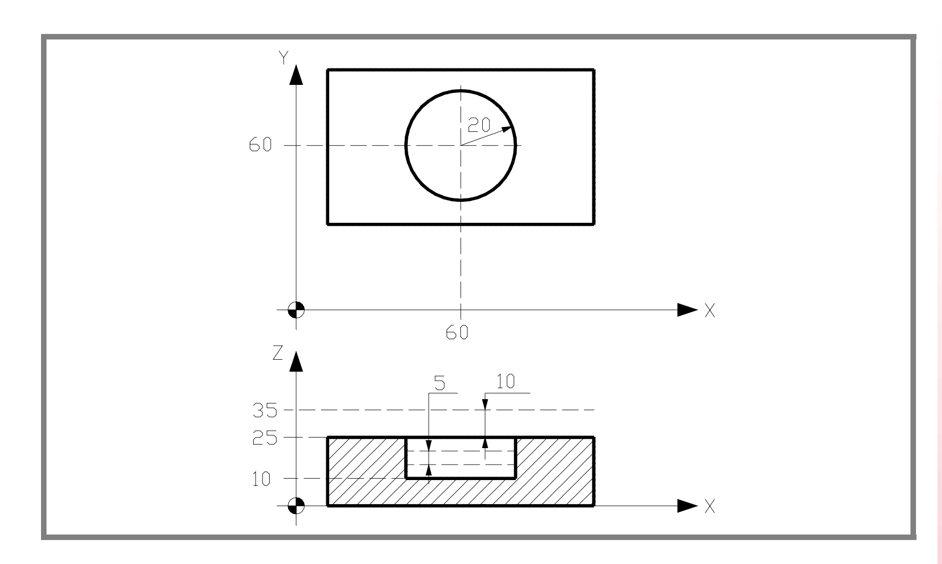
8. 进行下面的加工直到到达型腔的总深度。

- 切入工件，纵向轴以 "F" 指定的进给率切入工件至 "B" 确定的距离。
- 同步骤 5, 6 和 7 一样，铣削新的表面。

9. 如果功能 G98 有效，退回到初始平面 (Zi)；如果功能 G99 有效，退回到参考平面 (Z) 。

10.9.1 编程实例

加工一个半径为 20 mm 的型腔，圆心在 (X60, Y60)。型腔上表面是为 Z25，型腔加工到 Z10。参考平面设在 Z35。



G90 G0 X60 Y60
G88 Z35 I10 D10 J20 ·····

切深是 5 mm，以进给率 50 mm/min 执行。

G88 Z35 I10 D10 J20 **B5** ····· **V50**

粗加工路径铣削宽度是 5 mm，进给率 800 mm/min。在执行循环前必须选择铣削进给率，所以定义在前一程序段。

G90 G0 X60 Y60 **F800**
G88 Z35 I10 D10 J20 B5 **C5** ····· V50

保留精加工余量 1 mm，精加工将以 300 mm/min 的进给率执行。

G88 Z35 I10 D10 J20 B5 C5 **L1 H300** V50

现在说明如何执行型腔加工，加工在 (X200 Y135) 和 (X350 Y235) 多个点重复。

绝对方式编程 :

```
      T8 D1 M6
      G0 G90 X0 Y0 Z45 S1000 M3 M8 M41 F800
N10   G99 X60 Y60
      G88 Z35 I10 D10 J20 B5 C5 L1 H300 V50
N20   X200 Y135
N30   G98 X350 Y235
      M30
```




```
增量方式编程 :
      T8 D1 M6
      G0 G90 X0 Y0 Z45 S1000 M3 M8 M41 F800
N10   G99 G91 X60 Y60
      G87 Z-10 I-35 D10 J20 B5 C5 L1 H300 V50
N20   X140 Y75
N30   G98 X150 Y100
      M30
```

多重加工

11

编程者选择这种类型的加工，可以重复执行固定循环。

编程

用以下功能定义 :

G160	在直线模式上的多重加工
G161	在平行四边形模式上的多重加工
G162	在网格模式上的多重加工
G163	在整圆模式上的多重加工
G164	在圆弧模式上的多重加工
G165	在圆弧 - 弦模式上的多重加工

这些功能可以在任意工作平面执行。因为是非模态的所以每次使用时必须定义。

重复的机床操作必须是有效的。换句话说，这些功能仅在某个固定循环的影响下才有意义。

要完成多重加工按下列步骤进行：

1. 刀具运动到进行多重加工操作的第一点位置。
2. 定义在所有点重复的固定循环。
3. 定义要执行的多重加工操作。

注意事项

用这些指令编写的所有加工操作在相同的工作条件 (T, D, F, S) 下被执行，这些条件在定义固定循环时选择。

一旦执行了编写的多重加工，程序将恢复加工操作开始前的历史纪录，固定循环将保持有效。 F 现在是为固定循环编写的进给率。

同样地，刀具定位在执行编程加工操作的最后点的位置。

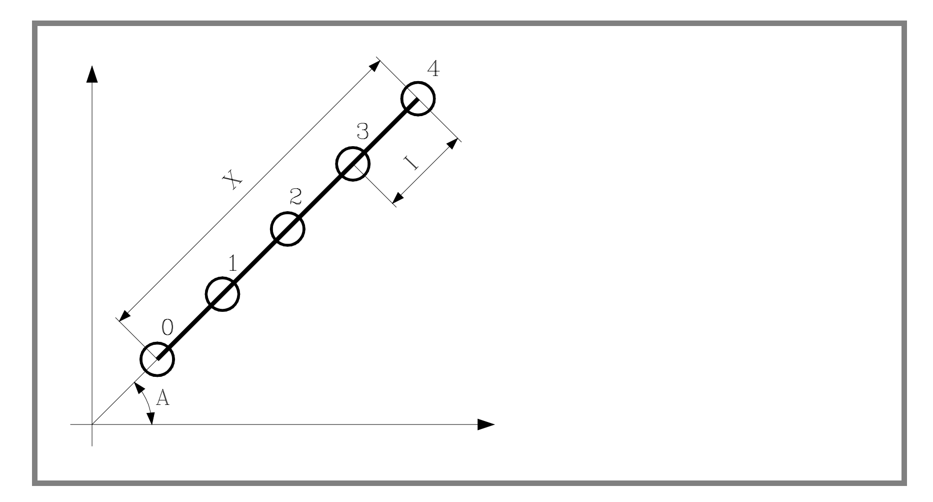
后述多重加工操作的详细介绍，均假定工作平面由 X 和 Y 轴构成。



11.1 G160. 在直线模式上的多重加工

该循环的编程格式如下：

G160	A	X I X K I K	P Q R S T U V



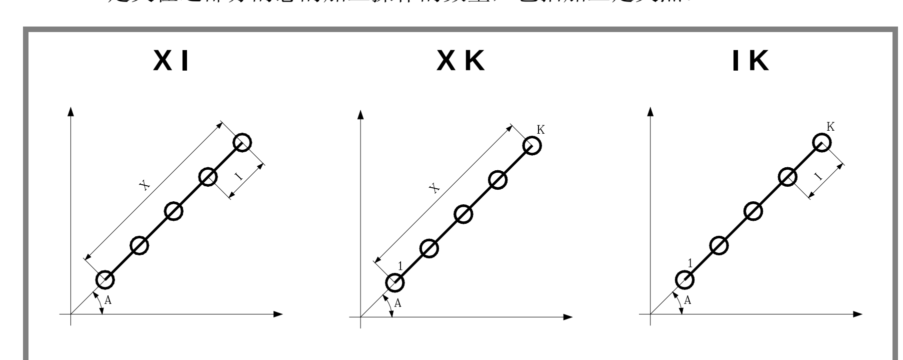
A　　定义加工操作路径和横坐标之间的夹角，单位为度。
　　　如果没有编写，采用 A = 0。

定义加工操作时，仅必需定义参数 "X", "I" 和 "K" 中的两个。

X　　定义加工路径的长度。

I　　定义加工操作之间的步长。

K　　定义在这部分的总的加工操作的数量，包括加工定义点。



当选择格式 "X-I"，必须注意保证加工操作数为整数，否则，数控系统将显示相应的错误代码。



P,Q,R,S,T,U,V　这些参数是可选的，用于指定在该点或在编写的这些点之间不需要进行加工。

因此，编写 "P7" 表示在点 7 不需要进行加工，编写 "Q10.013" 表示在点 10 到 13 不需要加工，也就是说在点 10, 11, 12 和 13 不需要加工。

当要定义一组点时 (Q10.013)，要注意用三位数字定义最终的点，如果编写了 "Q10.13"，多重加工理解为 "Q10.130"。

这些参数的编程顺序为 "P" "Q" "R" "S" "T" "U" "V"，赋予点的顺序也要遵循该顺序，即，赋予 "Q" 点的号必须大于赋予 "P" 小于赋予 "R" 的点的号。

例如 :

正确的编程	P5.006 Q12.015 R20.022
错误的编程	P5.006 Q20.022 R12.015

如果没有编写这些参数，数控系统理解为沿程序路径的所有点完成加工。

基本操作：

按照下述步骤执行多重加工：

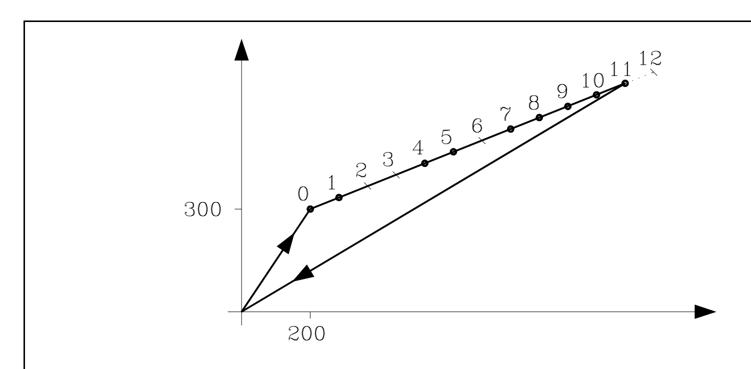
1. 多重加工计算编写的下一个将要加工的点。
2. 快速运动 (G00) 到该点。
3. 在该运动之后，多重加工将完成所选的固定循环。
4. 数控系统重复步骤 1-2-3，直到完成编写的多重加工操作 。

在完成多重加工之后，刀具将定位在完成加工的最后一点。



11.1.1 编程实例

假定工作平面由 X 和 Y 轴构成，纵向轴为 Z 轴，起始点为 X0 Y0 Z0：




```
G00 G91 X200 Y300 F100 S500
G98 G81 Z-8 I-22
G160 A30 X1200 I100 P2.003 Q6 R12
G80
G90 X0 Y0
M30
```

多重加工定义程序段也可以按下列方式编写：

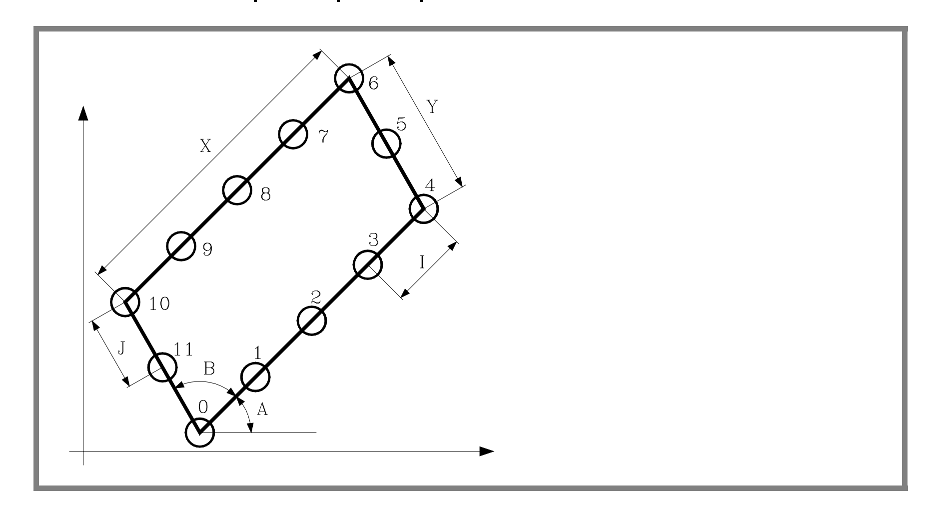
```
G160 A30 X1200 K13 P2.003 Q6 R12
G160 A30 I100 K13 P2.003 Q6 R12
```




11.2 G161. 在平行四边形模式上的多重加工

该循环的编程格式如下：

```
G161   A  B | X I | Y J | P Q R S T U V
            | X K | Y D |
            | I K | J D |
```



A　　定义加工操作路径和横坐标轴之间形成的夹角，单位为度。
　　　如果没有编写，采用 A = 0 。

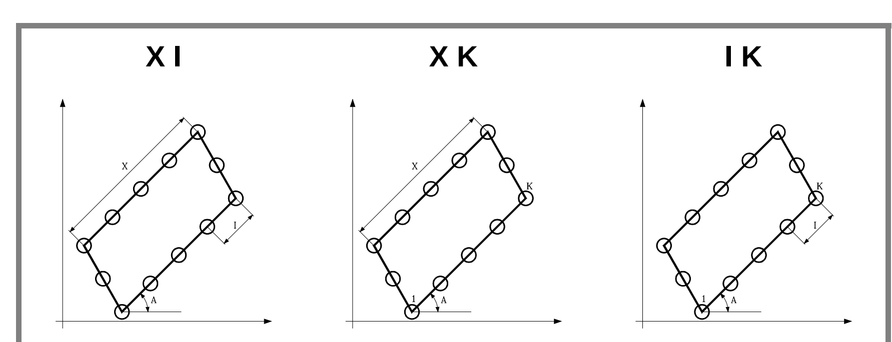
B　　定义两个加工路径形成的夹角，单位为度。
　　　如果没有编写，采用 B = 90。

当定义平行四边形的长度时，仅需定义参数 "X", "I" 和 "K" 其中的两个。

X　　定义沿横坐标轴方向平行四边形的长度。

I　　定义沿该路径加工操作之间的步长。

K　　定义沿该路径加工操作的总数量，包括加工定义点。



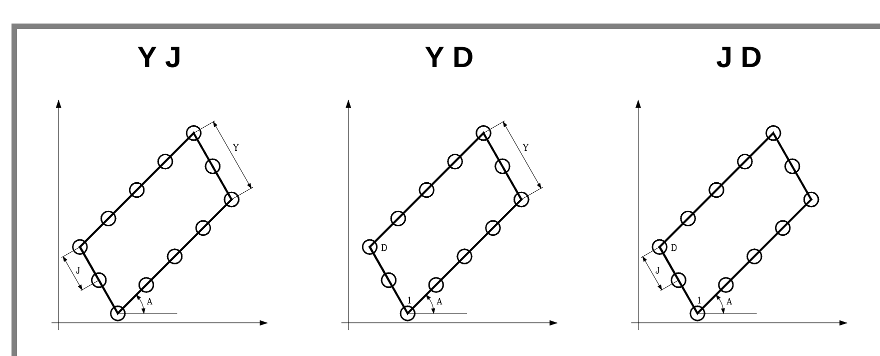
当选择格式 "X-I" 时 ，必须注意保证加工操作数为整数，否则，数控系统将显示相应的错误代码。

当定义平行四边形的宽度时，仅需编写 "Y", "J" 和 "D" 中的两个。

Y　　定义沿纵向轴方向的平行四边形的宽度。

J　　定义沿该方向加工操作之间的步长。

D　　定义沿该路径加工操作的总数量，包括加工定义点。



当选择格式 "Y-J" 时，必须注意保证加工操作数为整数，否则，数控系统将显示相应的错误代码。

P,Q,R,S,T,U,V　　这些参数是可选的，用于指定在该点或在编写的这些点之间不需要进行加工。

因此，编写 "P7" 表示在点 7 不需要进行加工，编写 "Q10.013" 表示在点 10 到 13 不需要加工，也就是说在点 10, 11, 12 和 13 不需要加工。

当要定义一组点时 (Q10.013)，要注意用三位数字定义最终的点，如果编写了 "Q10.13"，多重加工理解为 "Q10.130"。

这些参数的编程顺序为 "P" "Q" "R" "S" "T" "U" "V"，赋予点的顺序也要遵循该顺序，即，赋予 "Q" 点的号必须大于赋予 "P" 小于赋予 "R" 的点的号。

例如：

正确的编程	P5.006 Q12.015 R20.022
错误的编程	P5.006 Q20.022 R12.015

如果没有编写这些参数，数控系统理解为沿程序路径的所有点完成加工。

基本操作

按照下述步骤执行多重加工：

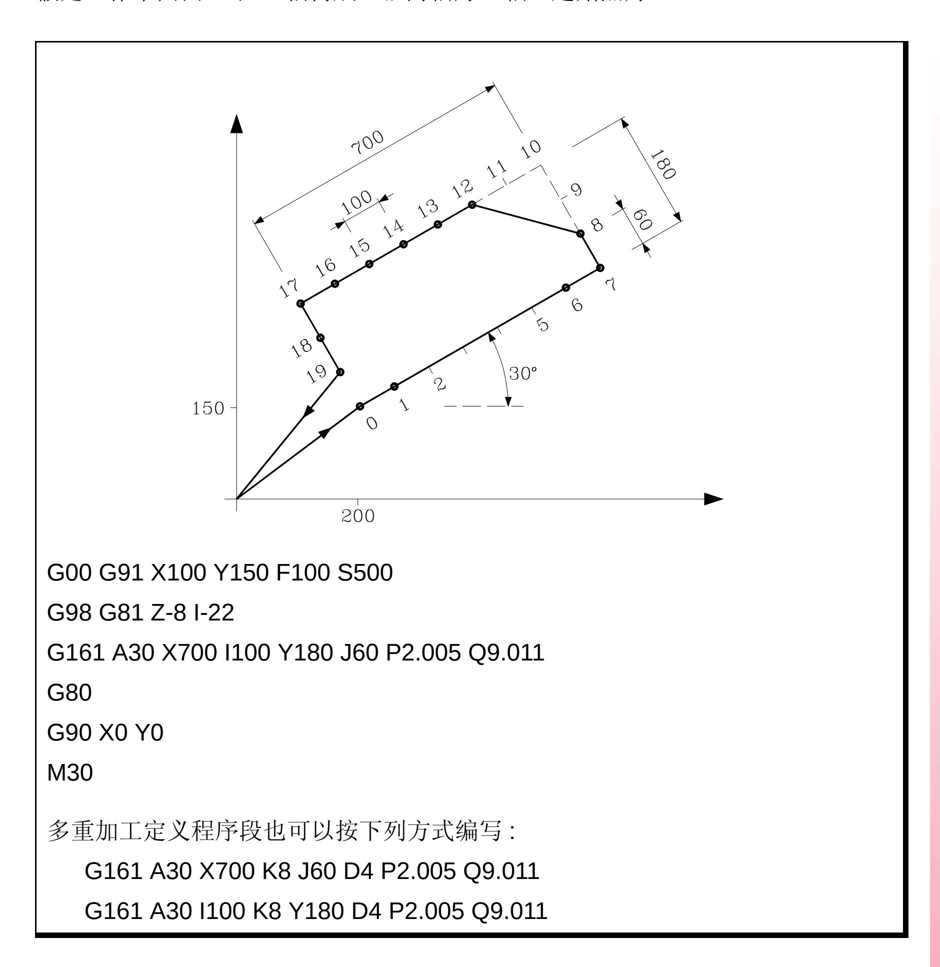
1. 多重加工计算编写的下一个将要加工的点。
2. 快速运动 (G00) 到该点。
3. 在该运动之后，多重加工将完成所选的固定循环。
4. 数控系统重复步骤 1-2-3，直到完成编写的多重加工操作 。

在完成多重加工之后，刀具将定位在完成加工的最后一点。



11.2.1 编程实例

假定工作平面由 X 和 Y 轴构成，纵向轴为 Z 轴，起始点为 X0 Y0 Z0：




```
G00 G91 X100 Y150 F100 S500
G98 G81 Z-8 I-22
G161 A30 X700 I100 Y180 J60 P2.005 Q9.011
G80
G90 X0 Y0
M30
```

多重加工定义程序段也可以按下列方式编写：

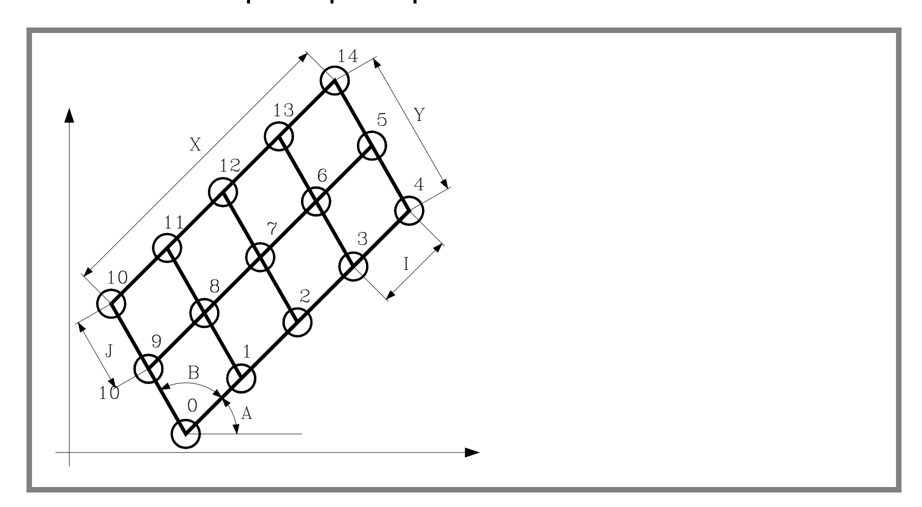
```
G161 A30 X700 K8 J60 D4 P2.005 Q9.011
G161 A30 I100 K8 Y180 D4 P2.005 Q9.011
```




11.3 G162. 在网格模式上的多重加工

该循环的编程格式如下：

G162 A B			
	X I	Y J	P Q R S T U V
	X K	Y D	
	I K	J D	



A　　定义加工操作路径和横坐标轴之间形成的夹角，单位为度。
　　如果没有编写，采用 A = 0 。

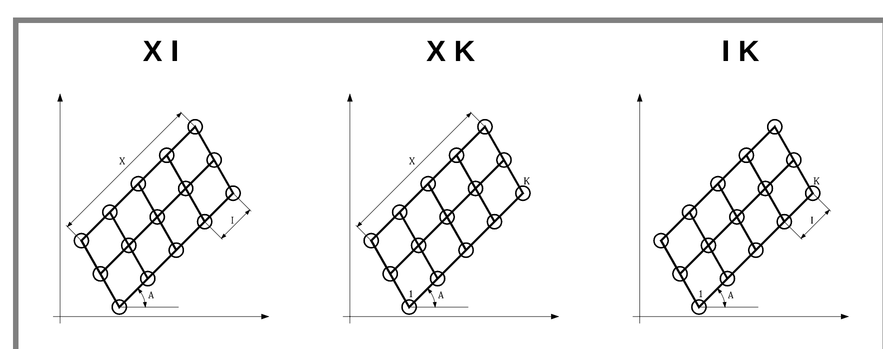
B　　定义两个加工路径形成的夹角，单位为度。
　　如果没有编写，采用 B = 90。

当定义网格的长度时，仅需定义参数 "X", "I" 和 "K" 中的两个。

X　　定义沿横坐标轴方向网格的长度。

I　　定义沿该路径加工操作之间的步长。

K　　定义沿该路径加工操作的总数量，包括加工定义点。



当选择格式 "X-I" 时 ，必须注意保证加工操作数为整数，否则，数控系统将显示相应的错误代码。

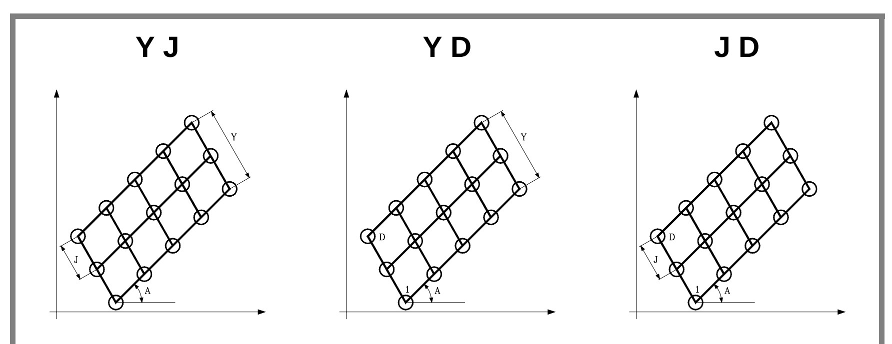


当定义网格的宽度时，仅必须编写 "Y", "J" 和 "D" 中的两个。

Y　　定义沿纵向轴方向的网格的宽度。

J　　定义沿该方向加工操作之间的步长。

D　　定义沿该路径加工操作的总数量，包括加工定义点。



当选择格式 "Y-J"， 必须注意保证加工操作数为整数，否则，数控系统将显示相应的错误代码。

P,Q,R,S,T,U,V　　这些参数是可选的，用于指定在该点或在编写的这些点之间不需要进行加工。

因此，编写 "P7" 表示在点 7 不需要进行加工，编写 "Q10.013" 表示在点 10 到 13 不需要加工，也就是说在点 10, 11, 12 和 13 不需要加工。

当要定义一组点时 (Q10.013)，要注意用三位数字定义最终的点，如果编写了 "Q10.13"，多重加工理解为 "Q10.130"。

这些参数的编程顺序为 "P" "Q" "R" "S" "T" "U" "V"，赋予点的顺序也要保证该顺序，即，赋予 "Q" 点的号必须大于赋予 "P" 小于赋予 "R" 的点的号。

例如 :

正确编程	P5.006 Q12.015 R20.022
错误编程	P5.006 Q20.022 R12.015

如果没有编写这些参数，数控系统理解为沿程序路径的所有点完成加工。

基本操作

按照下述步骤执行多重加工 :

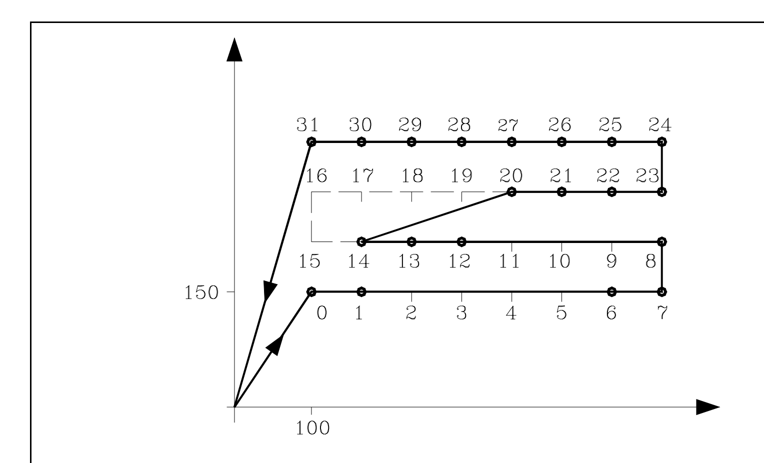
1. 多重加工计算编写的下一个将要加工的点。
2. 快速运动 (G00) 到该点。
3. 在该运动之后，多重加工将完成所选的固定循环。
4. 数控系统重复步骤 1-2-3，直到完成编写的多重加工操作 。

在完成多重加工之后，刀具将定位在完成加工的最后一点。



11.3.1 编程实例

假定工作平面由 X 和 Y 轴构成，纵向轴为 Z 轴，起始点为 X0 Y0 Z0：




```
G00 G91 X100 Y150 F100 S500
G98 G81 Z-8 I-22
G162 X700 I100 Y180 J60 P2.005 Q9.011 R15.019
G80
G90 X0 Y0
M30
```

多重加工定义程序段也可以按下列方式编写 :

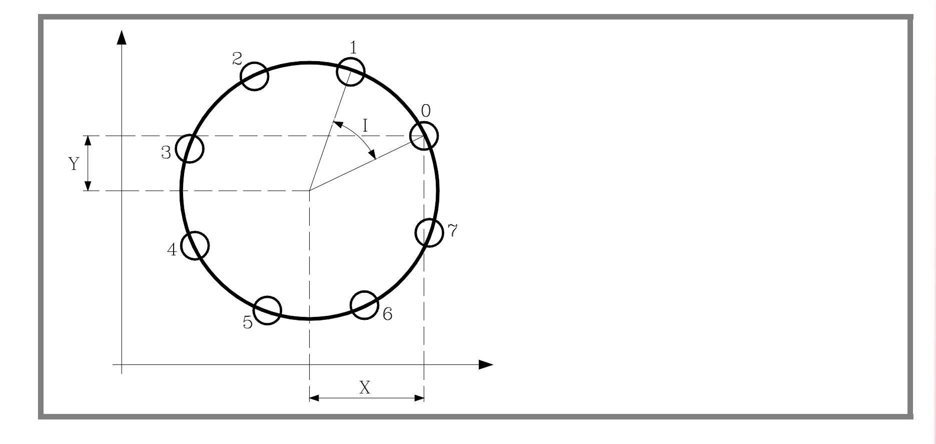
```
G162 X700 K8 J60 D4 P2.005 Q9.011 R15.019
G162 I100 K8 Y180 D4 P2.005 Q9.011 R15.019
```




11.4 G163. 在整圆模式上的多重加工

该循环的编程格式如下：

G163 X Y | I / K | C | F | P Q R S T U V



参数 "X" 和 "Y" 定义圆心，同在圆弧插补 (G02, G03) 中的 "I" 和 "J" 。

X　　定义沿横坐标从起点到圆心的距离。

Y　　定义沿纵坐标从起点到圆心的距离。

当定义加工操作，仅需定义参数 "I" 和 "K" 中的一个 。如果编写了角步长，记住总的角运动必须是 360º，否则，数控系统将发出相应的错误报告。

I　　定义加工操作之间的角步长。
当两点之间的运动以 G00 或 G01 进行，符号确定运动方向："I+" 逆时针方向， "I-" 顺时针方向。

K　　定义加工操作的总数量，包括加工定义点。
当点之间的运动以 G00 或 G01 进行，加工操作逆时针方向执行。

C　　定义在加工点之间的运动方式。如果没有编写，取值 C = 0 。

C=0　　快速定位 (G00)。

C=1　　直线插补 (G01)。

C=2　　顺时针圆弧插补 (G02)。

C=3　　逆时针圆弧插补 (G03)。

F　　加工点之间运动的进给率。仅在 "C" 值大于零时有效。



P,Q,R,S,T,U,V　这些参数是可选的，用于指定在该点或在编写的这些点之间不需要进行加工。

因此，编写 "P7" 表示在点 7 不需要进行加工，编写 "Q10.013" 表示在点 10 到 13 不需要加工，也就是说在点 10, 11, 12 和 13 不需要加工。

当要定义一组点时 (Q10.013)，要注意用三位数字定义最终的点，如果编写了 "Q10.13"，多重加工理解为 "Q10.130"。

这些参数的编程顺序为 "P" "Q" "R" "S" "T" "U" "V"，赋予点的顺序也要保证该顺序，即，赋予 "Q" 点的号必须大于赋予 "P" 小于赋予 "R" 的点的号。

例如 :

正确编程	P5.006 Q12.015 R20.022
错误编程	P5.006 Q20.022 R12.015

如果没有编写这些参数，数控系统理解为沿程序路径的所有点完成加工。

基本操作

按照下述步骤执行多重加工 :

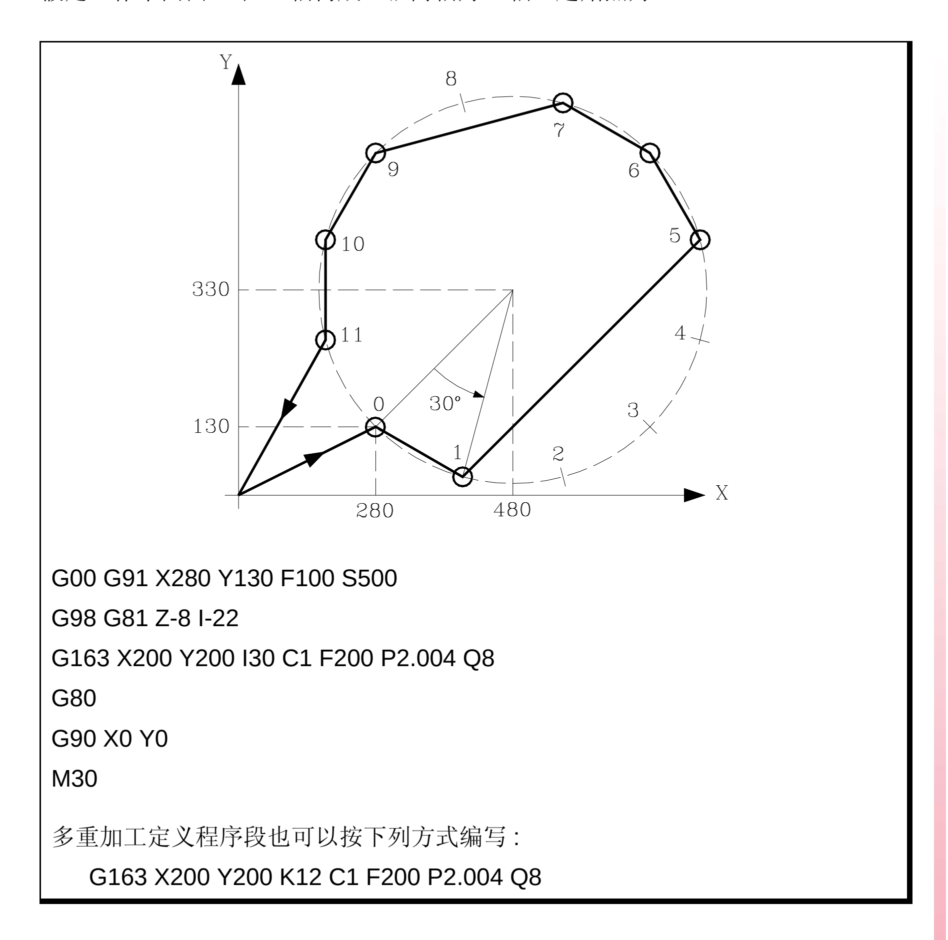
1. 多重加工计算编写的下一个将要加工的点。
2. 以 "C" 编写的方式 (G00, G01, G02 或 G03) 运动到该点。
3. 在该运动之后，多重加工将完成所选的固定循环。
4. 数控系统重复步骤 1-2-3，直到完成编写的多重加工操作 。

在完成多重加工之后，刀具将定位在完成加工的最后一点。



11.4.1 编程实例

假定工作平面由 X 和 Y 轴构成，纵向轴为 Z 轴，起始点为 X0 Y0 Z0：




```
G00 G91 X280 Y130 F100 S500
G98 G81 Z-8 I-22
G163 X200 Y200 I30 C1 F200 P2.004 Q8
G80
G90 X0 Y0
M30
```

多重加工定义程序段也可以按下列方式编写 :

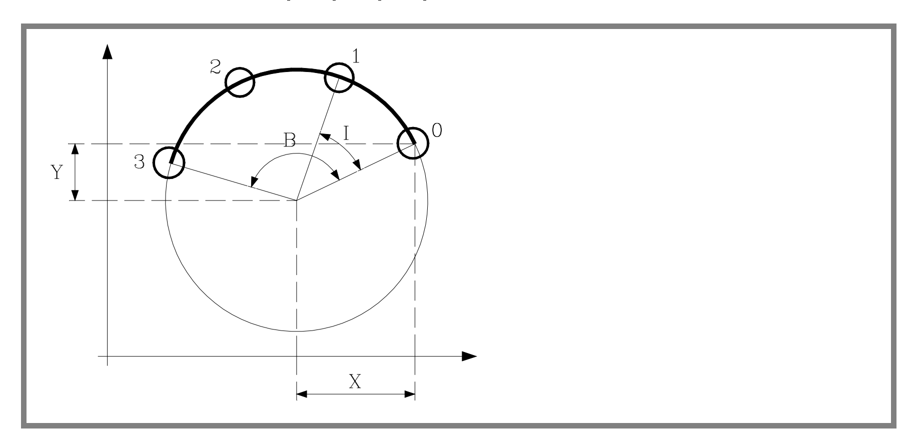
```
G163 X200 Y200 K12 C1 F200 P2.004 Q8
```




11.5 G164. 在圆弧模式上的多重加工

该循环的编程格式如下：

G164 X Y B | I / K | C | F | P Q R S T U V



参数 "X" 和 "Y" 定义圆心，同在圆弧插补 (G02, G03) 中 "I" 和 "J" 。

X 定义沿横坐标从起点到圆心的距离。

Y 定义沿纵坐标从起点到圆心的距离。

B 定义加工路径的角距离。

当定义加工操作，仅必需定义参数 "I" 和 "K" 中的一个 。如果编写了角步长，记住总的角运动必须是 "B" 确定的角距离，否则，数控系统将发出相应的错误报告。

I 定义加工操作之间的角步长。
当两点之间的运动以 G00 或 G01 进行，符号确定运动方向："I+" 逆时针方向， "I-" 顺时针方向。

K 定义加工操作的总数量，包括加工定义点。
当点之间的运动以 G00 或 G01 进行，加工操作逆时针方向执行。

C 定义在加工点之间的运动方式。如果没有编写，取值 C = 0 。

C=0 快速定位 (G00).

C=1 直线插补 (G01).

C=2 顺时针圆弧插补 (G02).

C=3 逆时针圆弧插补 (G03).

F 加工点之间运动的进给率。仅在 "C" 值大于零时有效。



P,Q,R,S,T,U,V　这些参数是可选的，用于指定在该点或在编写的这些点之间不需要进行加工。

因此，编写 "P7" 表示在点 7 不需要进行加工，编写 "Q10.013" 表示在点 10 到 13 不需要加工，也就是说在点 10, 11, 12 和 13 不需要加工。

当要定义一组点时 (Q10.013)，要注意用三位数字定义最终的点，如果编写了 "Q10.13"，多重加工理解为 "Q10.130"。

这些参数的编程顺序为 "P" "Q" "R" "S" "T" "U" "V"，赋予点的顺序也要遵循该顺序，即，赋予 "Q" 点的号必须大于赋予 "P" 小于赋予 "R" 的点的号。

例如 :

正确编程	P5.006 Q12.015 R20.022
错误编程	P5.006 Q20.022 R12.015

如果没有编写这些参数，数控系统理解为沿程序路径的所有点完成加工。

基本操作

按照下述步骤执行多重加工 :

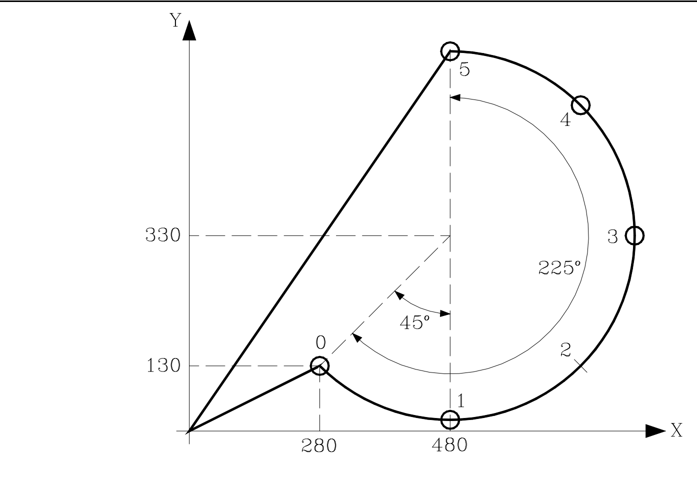
1. 多重加工计算编写的下一个将要加工的点。
2. 以 "C" 编写的方式 (G00, G01, G02 或 G03) 运动到该点。
3. 在该运动之后，多重加工将完成所选的固定循环。
4. 数控系统重复步骤 1-2-3，直到完成编写的多重加工操作 。

   在完成多重加工之后，刀具将定位在完成加工的最后一点。



11.5.1 编程实例

假定工作平面由 X 和 Y 轴构成，纵向轴为 Z 轴，起始点为 X0 Y0 Z0：




```
G00 G91 X280 Y130 F100 S500
G98 G81 Z-8 I-22
G164 X200 Y200 B225 I45 C3 F200 P2
G80
G90 X0 Y0
M30
```

多重加工定义程序段也可以按下列方式编写 :

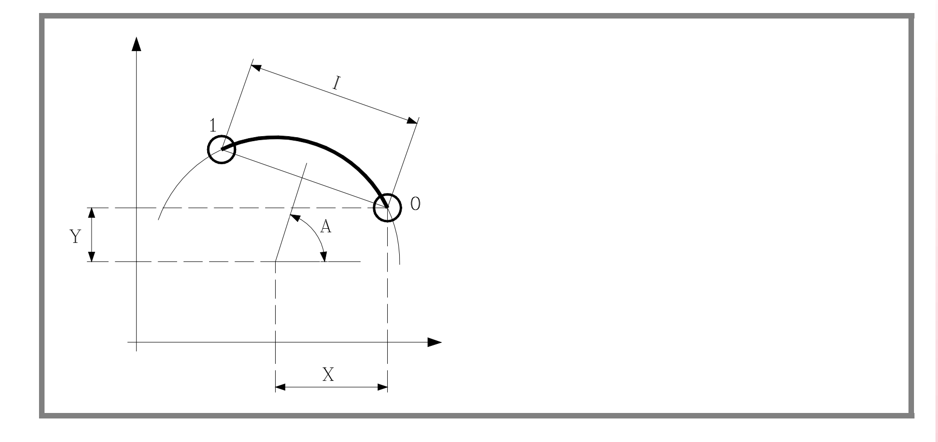
```
G164 X200 Y200 B225 K6 C3 F200 P2
```




11.6 G165. 在弦模式上的多重加工

该功能可以激活通过弦编写的点的加工操作。仅完成一个加工操作，编程格式如下：

```
G165   X | A | C | F
           I
```



参数 "X" 和 "Y" 定义圆心，同在圆弧插补 (G02, G03) 中的 "I" 和 "J" 。

X　　定义沿横坐标从起点到圆心的距离。

Y　　定义沿纵坐标从起点到圆心的距离。

当定义加工操作时，仅必需定义参数 "A" 和 "I" 中的一个 。

A　　定义弦的垂直平分线和横坐标轴形成的角度，单位为度。

I　　定义弦的长度。
当两点之间的运动以 G00 或 G01 进行，符号确定运动方向："I+" 逆时针方向， "I-" 顺时针方向。

C　　定义在加工点之间的运动方式。如果没有编写，取值 C = 0 。

C=0　　快速定位 (G00).

C=1　　直线插补 (G01).

C=2　　顺时针圆弧 (G02).

C=3　　逆时针圆弧 (G03).

F　　加工点之间运动的进给率。仅在 "C" 值大于零时有效。

基本操作

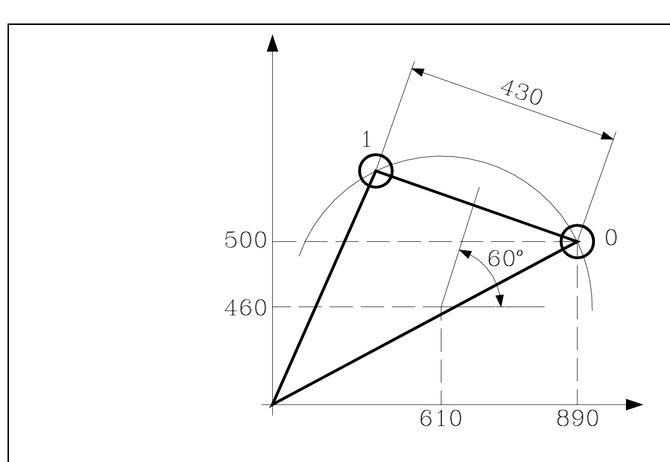
按照下述步骤执行多重加工 :

1. 多重加工计算编写的下一个将要加工的点。
2. 以 "C" 编写的方式 (G00, G01, G02 或 G03) 运动到该点。
3. 在该运动之后，多重加工将完成所选的固定循环。

   在完成多重加工之后，刀具将定位在完成加工的最后一点。

11.6.1 编程实例

假定工作平面由 X 和 Y 轴构成，纵向轴为 Z 轴，起始点为 X0 Y0 Z0：




```
G00 G91 X890 Y500 F100 S500
G98 G81 Z-8 I-22
G165 X-280 Y-40 A60 C1 F200
G80
G90 X0 Y0
M30
```

多重加工定义程序段也可以按下列方式编写 :

```
G165 X-280 Y-40 I430 C1 F200
```

循环编辑

12

12.1 概述

集成至循环编辑里面的循环可分为以下几类：

加工固定循环

- 钻孔

  中心钻，钻孔 1，钻孔 2。
- 攻螺纹
- 铰孔
- 镗孔

  镗孔 1，镗孔 2。
- 型腔

  型腔 : 简单，矩形，圆形，预制空腔，2D， 3D。
- 凸台

  凸台：矩形，圆形。
- 平面铣削
- 轮廓铣削

  点到点轮廓 , 轮廓
- 狭槽铣削

多重加工

- 直线
- 圆弧
- 矩形
- 网格
- 任意 (用户定义的多个点).

多重加工可与固定循环关联，所以可以在多个点重复操作。



执行

当执行这些固定循环时，数控系统在表格中列出如下可用的 "G" 功能。

G281	中心钻
G282	钻孔 1
G283	钻孔 2
G284	攻螺纹
G285	铰孔
G286	镗孔 1
G297	镗孔 2
G287	矩形型腔
G288	圆形型腔
G289	简单型腔
G296	预制空腔型腔
G291	矩形凸台
G292	圆形凸台
G290	平面铣削
G293	点到点轮廓
G294	轮廓
G295	狭槽铣削



12.1.1 关联固定循环的多重加工操作

多重加工可以和下列循环关联：

- 中心钻，钻孔 1, 钻孔 2, 攻螺纹，铰孔，镗孔 1, 镗孔 2。
- 简单，矩形，圆形和预制空腔型腔。
- 矩形和圆形凸台。

可是，没有多重加工可以和下列循环关联：

- 2D 和 3D 型腔，平面铣削，轮廓铣削，点到点轮廓和狭槽铣削。

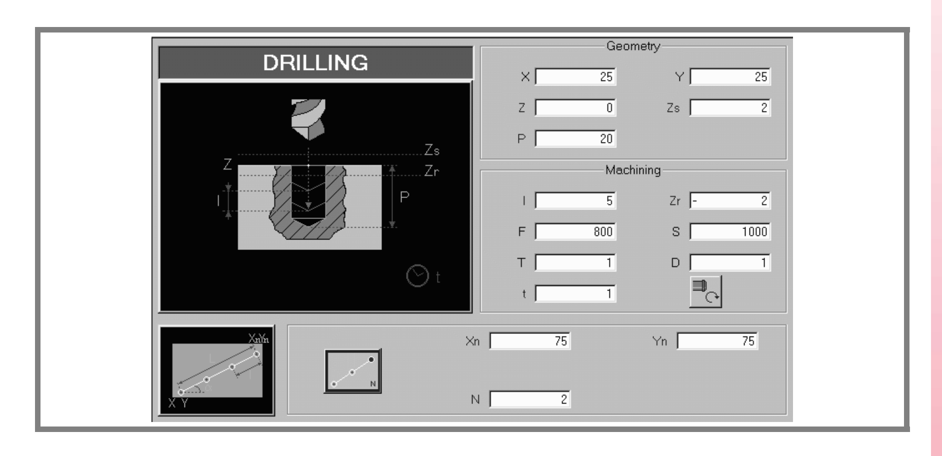
关联固定循环到多重加工操作的做法：

1. 选择和定义固定循环。

2. 按下 “多重” 软键。

3. 选择要求的多重加工操作。

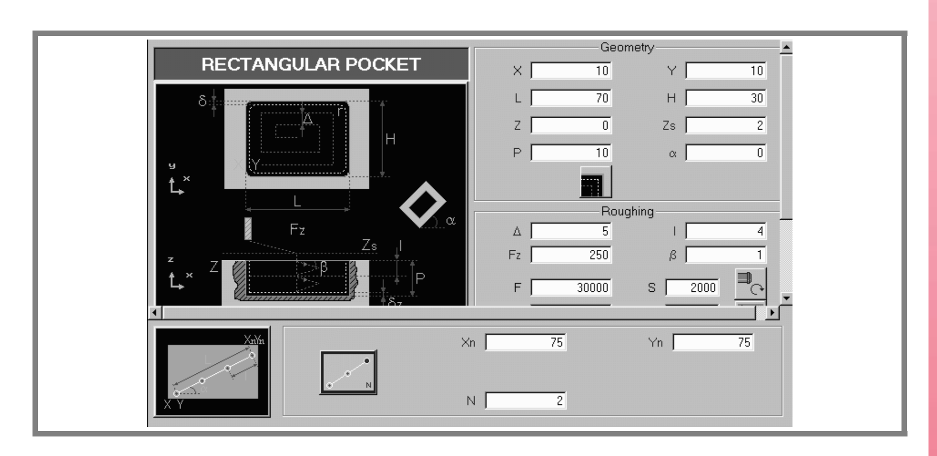
下图所示，与钻孔 1 循环关联的直线模式上的多重加工操作。



(a)

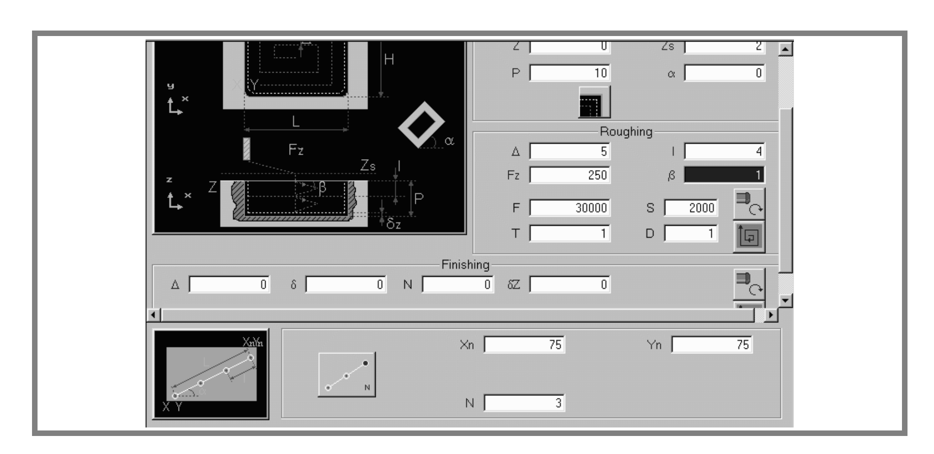
要编辑固定循环或多重加工的数据，用按键 (a) 选择相应的窗口。

当固定循环占据整个屏幕时，多重加工操作在上层运行，如图所示。





在这些情况下，编辑循环数据时，最上层的窗口是可以为显示数据而自动移动。





固定循环编辑窗口是通用的。不从属于有效的工作平面。

固定循环没有和工件平面关联在一起，它们在当前有效平面执行。

使用与 G17 工作平面相同的命名法。

X 横坐标轴。

Y 纵坐标轴。

Z 垂直轴 .

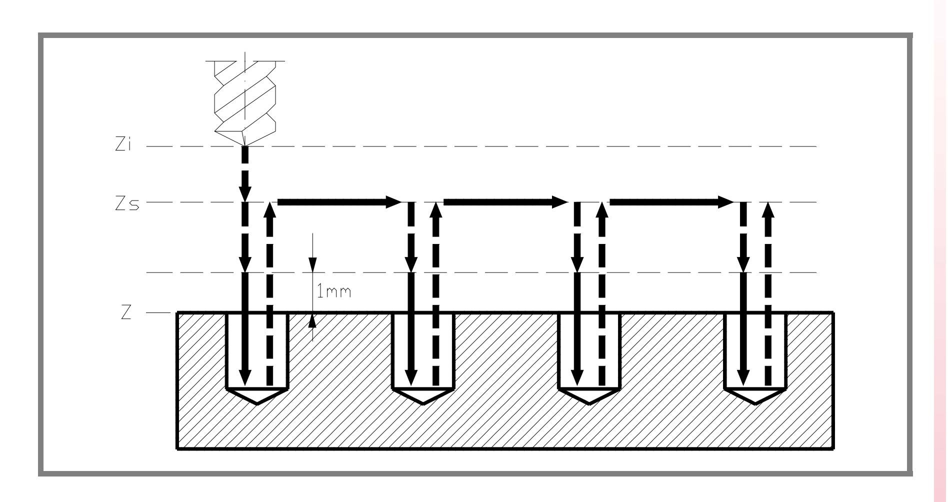
当在另一个平面工作时，必须：

- *选择正确的工作平面。*
  G17, G18, G19 或指令 #SET AX。
- *选择纵向轴和加工方向。*
  指令 #TOOL AX。
- *考虑先前的命名法，来编写固定循环。*

12.1.2 加工运动

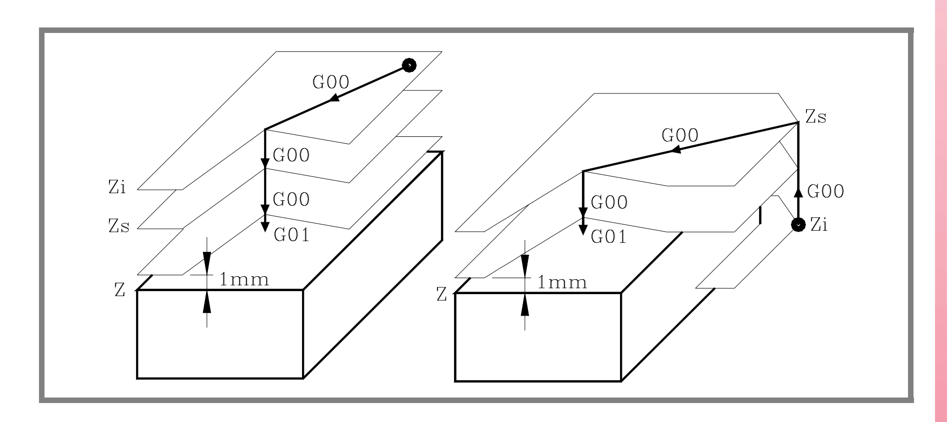
在所有的操作中有四个工作平面：

- 初始平面或调用循环时刀具位置 Zi。不需要定义它 。
- 安全平面。用于首次接近和在加工操作间移动刀具。用循环参数 Zs 定义。
- 接近平面。不需要定义。由数控系统计算，离工件表面 1 mm。
- 工件表面。用参数 Z 定义。



当执行循环时，刀具快速 (G0) 运动到安全平面 (Zs)：

- 如果初始平面在安全平面以上 （左图），先在 X 和 Y 轴运动，然后是 Z 轴。
- 如果初始平面在安全平面以下 （右图），先在 Z 轴运动，然后是 X 和 Y 轴。



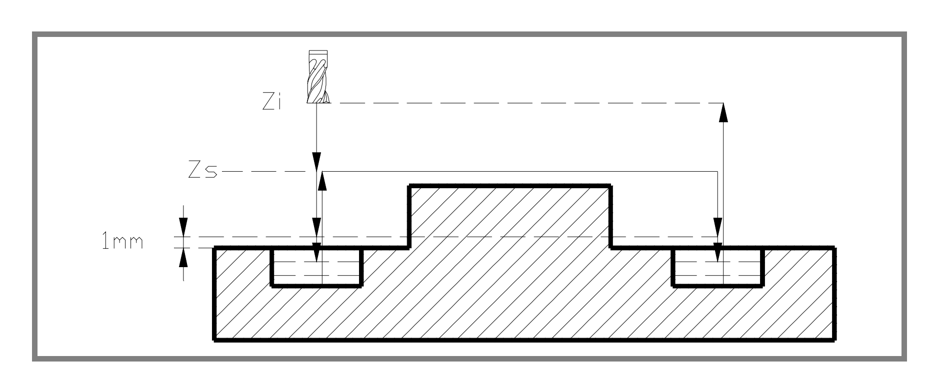
然后，快速 (G0) 运动到接近平面，最后以工进速度执行加工操作。

一旦加工操作结束，刀具返回到安全平面 (Zs)。



如果有和它相关联的多重加工，刀具在 XY 运动，沿安全平面 (Zs)，直到下一个加工点。

在如图所示的情况下，当安全平面（Zs）远离工件表面时，允许快速接近加工表面。



12.1.3 数据，轮廓，图标的选择

数据选择

输入或修改数据，必须选择；也就是，必须进行编辑。

固定循环的参数可以用 [🡄] [🡆] [🡅] [🡇] 键选择或按直接存储键。每一组的第一个数据也可以按 page-up （上翻页）和 page-down（下翻页）键来选择。

直接存储键对应参数名称； [F] 为进给率 , [T] 为刀具 , 等等 。每一次按下相同的键，选择相同类型的下一个数据。

数据输入

放置指针到相应的窗口，键入期望值并按下 [ENTER] 键。如果没有按下 [ENTER] 键，不会采用新值。

如果选择 Teach-in （示教）模式，机床当前位置可以和坐标关联。放置指针到相应的窗口，按下 [RECALL] 键。

对于 X 轴参数，将当作该通道的第一轴坐标，在编辑模拟模式有效的地方。对于 Y 轴参数，将当作该通道的第二轴的坐标。对于 Z 轴参数，将当作该通道的第三轴坐标。

变换图标状态

放置指针到期望的图标，接着按空格键。

选择 - 定义轮廓

放置指针到相应的窗口。

选择轮廓，按 [🡇] 键来扩展定义轮廓的菜单，然后选择一个或键入它的名称。

定义新的轮廓，键入希望使用的名称或按 [RECALL] 键。访问轮廓编辑。

修改现有的轮廓，键入名称或按 [RECALL] 键。访问轮廓编辑。

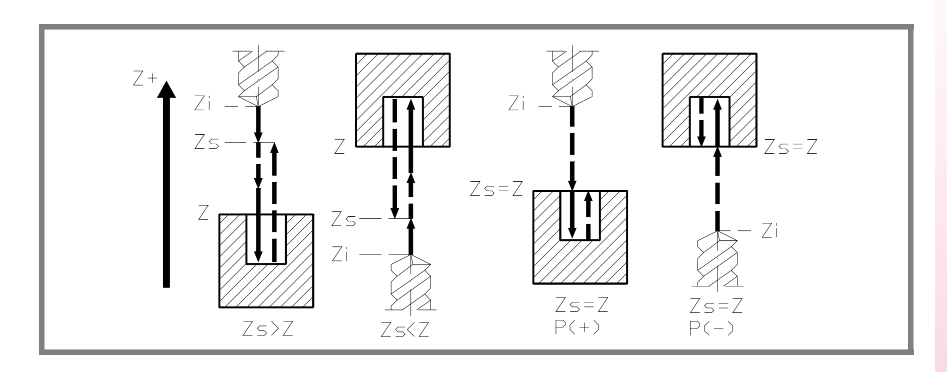
12.1.4 当参数值为 0 时应用的值

加工方向：

Z 和 Zs 设置加工方向。

如果 Z=Zs, 方向由 P （总深度）的符号设置。

如果 P(+) 方向朝向 Z(-)， 如果 P(-) 方向朝向 Z(+)。



穿透步长 I=0:

当编写 I=0 时，采用刀具表中赋给刀具的切削长度作为步长。

如果刀具表中的值也为 0，将导致错误。

穿透进给率 Fz=0:

当编写 Fz=0 时，粗加工和精加工切入以为每个操作选择的铣削进给率 "F" 的一半来执行 。

穿透角 β=0 和 θ=0:

在这两种情况时，当编写 0 时，采用刀具表中的赋值 。

如果刀具表中的值也是 0, 则直接穿透 （90° 角），没有倾斜角 。

精加工走刀数或穿透次数 N=0:

当编写 N=0 时，执行可能最少的走刀次数，考虑在刀具表中的刀具的长度值。

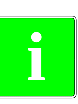
在型腔和凸台 (除了 2D 和 3D 型腔) 中，如果刀具表中的值是 0，检查粗加工和精加工刀具。如果没有变化，在粗加工操作之后，精加工执行切入和退出。

如果它们不相等，发送出错信息。



12.1.5 模拟固定循环

在固定循环编辑器中，可以模拟编辑的固定循环，不需要模拟整个零件程序。在模拟的时候，可以察看和编辑另一个固定循环，并且可以返回程序编辑。

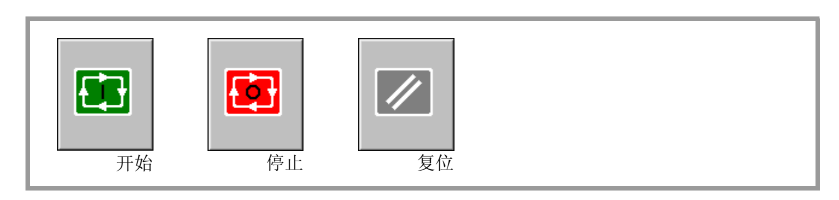


如果循环编辑包含在自动操作模式里，不可以模拟循环。

模拟循环

按下 [START] 图标 开始模拟正在编辑的循环。模拟可以用 [STOP] 图标中断或用 [RESET] 图标取消。

模拟图形总是叠加在主循环的帮助图形之上。如果循环有与其相关联的定位，图形叠加在主循环之上；在 2D 型腔有钻孔的情况时。



一旦模拟开始，继续到循环结束或按下 [RESET] 图标。甚至在模拟期间变换循环或返回程序编辑时，在前的循环在模拟期间仍有效。

循环模拟窗口

图形窗口 （在模拟时）在按下 [START] 图标时被激活，在按下 [RESET] 图标时被取消。此窗口放置在循环帮助图形之上；可以用组合键 [CTRL]+[G］扩大到全屏（或再次缩小）。

窗口的左下角显示循环的名字和模拟通道，即调用循环编辑器的程序编辑器通道。

设置图形环境

当激活或选择图形窗口时，水平的软键菜单显示可用的图形选项。相关图形选项的详细信息，请参阅操作手册有关编辑 – 模拟模式的章节。

一些图形选择也可以手工编辑。编辑区域仅在窗口扩大 ([CTRL]+[G]) 时进行显示。

模拟图形一直保持有效，直到被清除掉；也就是开始模拟新的循环并不清除先前的图形。



显示图形的最佳区域

显示区域可以从和模拟图形窗口相关的软键菜单来确定，或可以留给数控系统周期性的计算最佳的区域。

当图形窗口可视时，组合键 [CTRL]+[D] 激活最佳区域计算。从那时起直到停止循环编辑，数控系统周期性的计算显示图形的最佳区域。

当退出图形时，将采用最后一次的计算结果作为新的显示区域。

模拟和数据编辑的窗口

选择图形窗口时，可以用直接访问键切换循环参数区域。如果参数属于定位循环，按下 [CTRL]+[F2](窗口切换) 。

如果循环是在全屏模拟，循环编辑也可通过按 [ESC] 键访问。再次选择图形窗口，用组合键 [CTRL]+[G] 或 [SHIFT]+[G] 或 [G]。

当图形窗口聚焦或者循环编辑时，水平的软键菜单将显示图形选项。

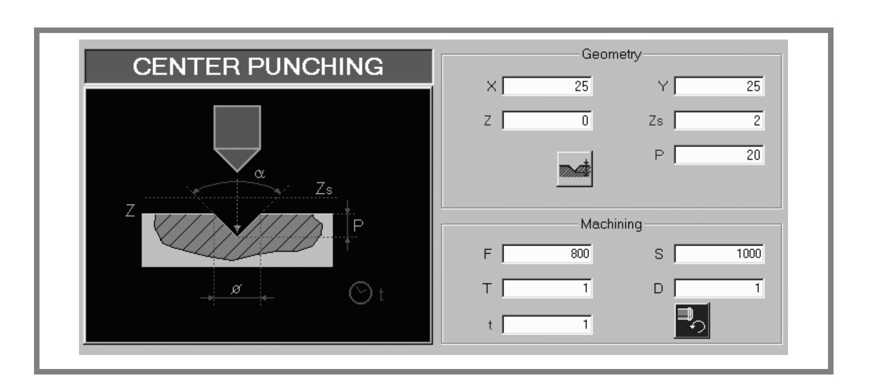
当编辑数据时，正在进行的模拟不会中断。如果在模拟过程中改变循环数据，将会在下次循环模拟中采用；也就是一旦完成或在 STOP 和 RESET 中途中断之后会重新安排模拟。

模拟循环时的热键汇总

[CTRL]+[F2]	在参数窗口，拴牢循环参数和定位参数。
[CTRL]+[G]	选择图形窗口。 缩小或放大图形窗口。 为图形数据显示对话区域。
[CTRL]+[D]	激活最佳区域周期性计算。
[SHIFT]+[G] [G]	当模拟正在运行时显示图形窗口，参数编辑窗口是有效的。
[ESC]	如果图形以全屏显示，它显示循环编辑屏幕。



12.2 中心钻

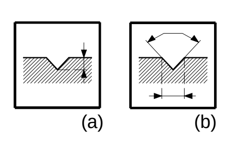


几何参数 :

X, Y　加工点

Z　工件表面坐标

Zs　安全平面坐标


(a) (b)

编程类型深度 (图标)。

P　总深度。使用图标 (a)。

τo　中心钻角度。使用图标 (b)。

ϕ　中心钻直径。使用图标 (b)。

使用 Z=Zs 和图标 (b)，加工方向总是朝向 Z(-)。

加工参数 :

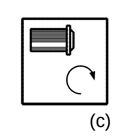
F　进给率。

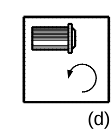
S　主轴速度。

T　刀具。

D　刀具偏置。

t　底面暂停，单位为秒。


(c)


(d)

主轴旋转方向 (图标).

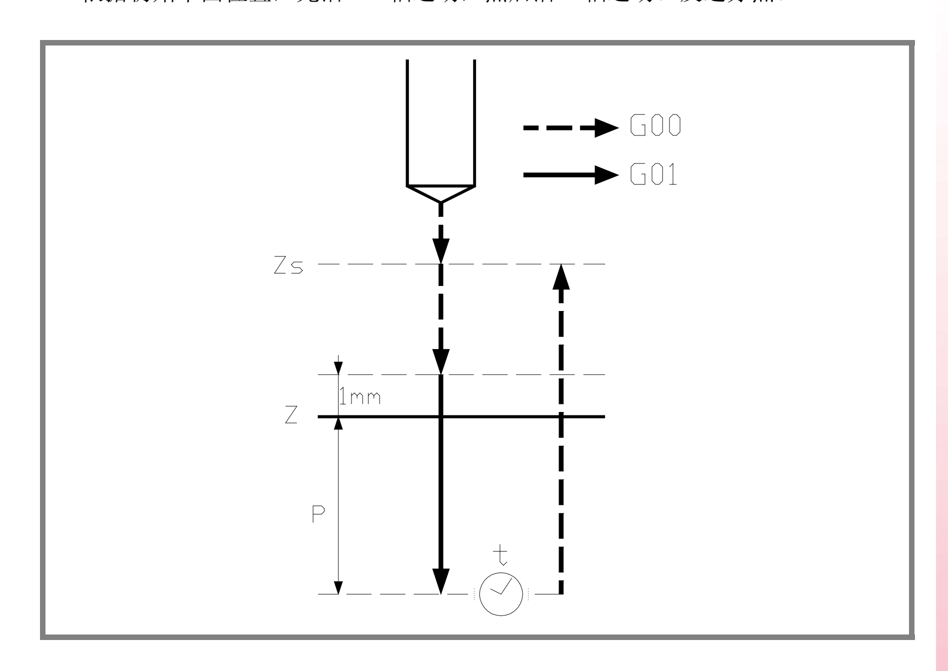
顺时针使用图标 (c) 和逆时针使用图标 d)

12.2.1 基本操作：

1. 启动主轴以需要的方向旋转。

2. 快速运动 (G0)，直到 XY 点和安全平面 (Zs).

依据初始平面位置，先沿 XY 轴运动，然后沿 Z 轴运动。反之亦然。



3. 快速 (G0) 运动到接近平面。

4. 以进给率 "F" 进刀。

5. 暂停 "t"。

6. 快速返回 (G0) 到安全平面 (Zs)。

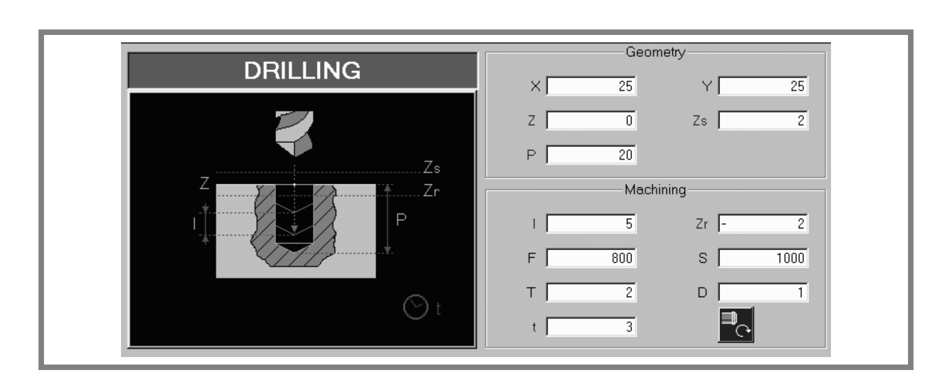
如果关联多重加工操作，将尽可能频繁的执行下列步骤：

7. 快速 (G0) 运动到下一点。

8. 重复步骤 3, 4, 5, 6。



12.3 钻孔 1



几何参数 :

X, Y 加工点。

Z 工件表面坐标。

Zs 安全平面坐标。

P 总深度。

加工参数 :

I 单步进刀深度。钻孔以给定步长进行，除了最后一步是加工余量。

Zr 每次单步钻孔后快速 (G0) 返回的替代坐标位置。

如果没有到达 "Zr" 坐标 , 返回接近平面。

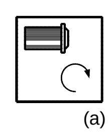
F 进给率。

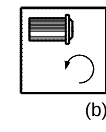
S 主轴速度。

T 刀具。

D 刀具偏置。

t 底面暂停，单位为秒。


(a)

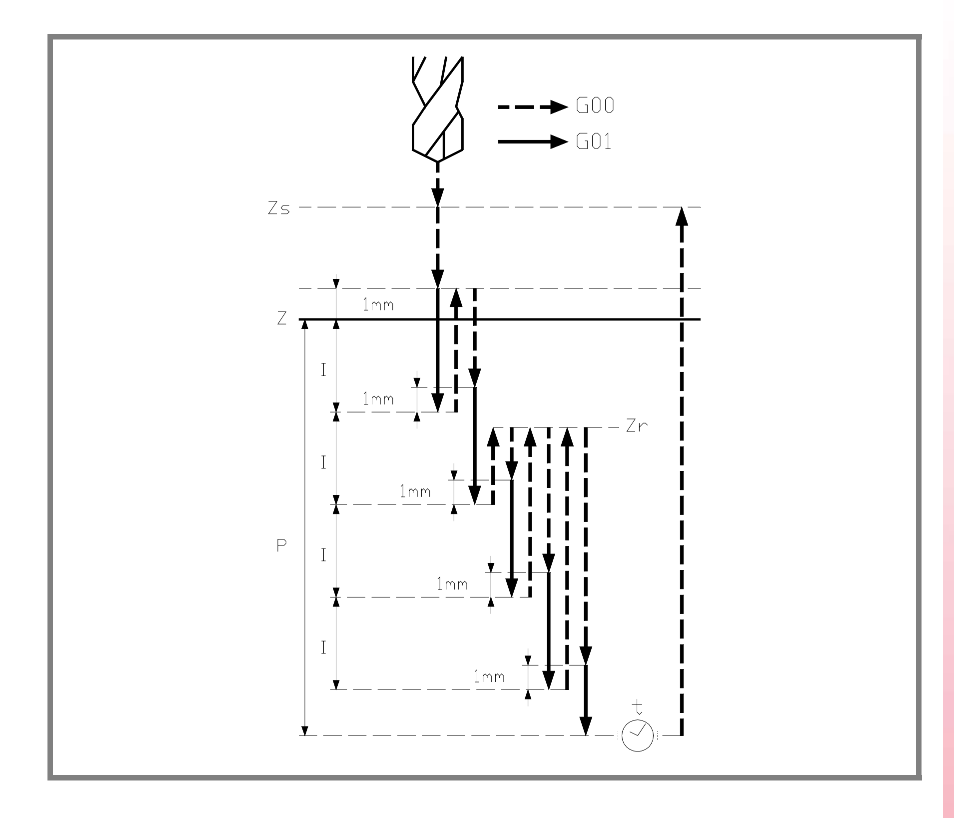

(b)

主轴旋转方向 (图标).

顺时针使用图标 (a) 和 逆时针使用图标 (b)

12.3.1 基本操作：

1. 启动主轴以需要的方向旋转。
2. 快速运动(G0)，直到XY点和安全平面(Zs)。依据初始平面位置，先沿XY轴运动，然后是沿 Z 轴运动。反之亦然。

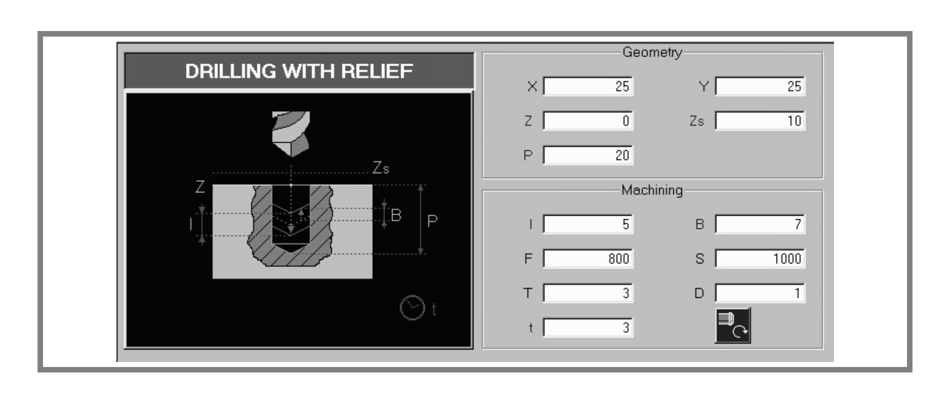


3. 快速 (G0) 运动到接近平面。
4. 以进给率 "F" 进刀 "I" 距离。
5. 钻削循环直到到达总的深度 "P"。
   ·1· 快速返回 (G0) 到替代坐标 Zr。
   如果没有到达 "Zr" 坐标，返回接近平面。
   ·2· 快速接近 (G0) 直到距前次单步钻削到位置 1 mm 处 。
   ·3· 以进给率 "F" 进刀 "I" 距离。
6. 暂停 "t" 秒。
7. 快速退回 (G0) 到安全平面 (Zs)。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

8. 快速运动 (G0) 到下一点 。
9. 钻削新孔，重复步骤 3, 4, 5, 6, 7。

12.4 钻削 2



几何参数 :

X, Y 加工点。

Z 工件表面坐标。

Zs 安全平面坐标。

P 总深度。

加工参数 :

I 单步进刀深度。钻孔以给定步长进行，除了最后一步是加工余量。

B 每次单步钻孔后快速 (G0) 返回的替代坐标位置。

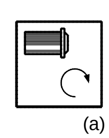
F 进给率。

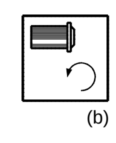
S 主轴速度。

T 刀具。

D 刀具偏置。

t 底面暂停，单位为秒。


(a)

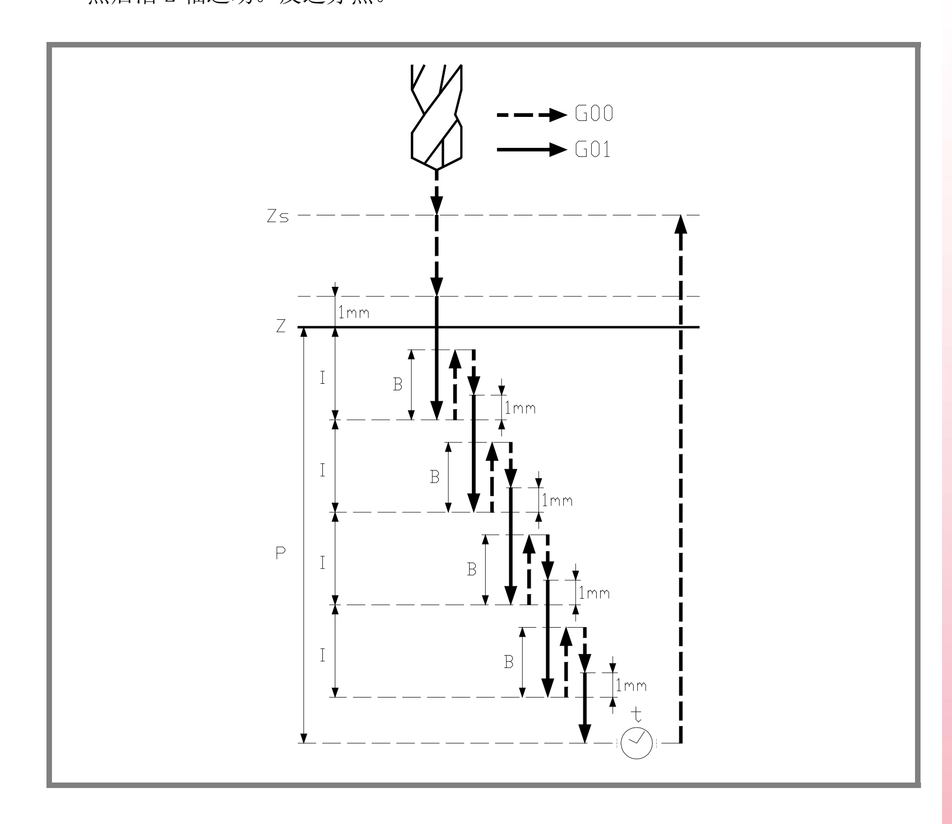

(b)

主轴旋转方向 (图标).

顺时针使用图标 (a) 和逆时针使用图标 (b)

12.4.1 基本操作：

1. 启动主轴以需要的方向旋转。
2. 快速运动 (G0), 直到 XY 点和安全平面 (Zs)。依据初始平面位置，先沿 XY 轴运动，然后沿 Z 轴运动。反之亦然。



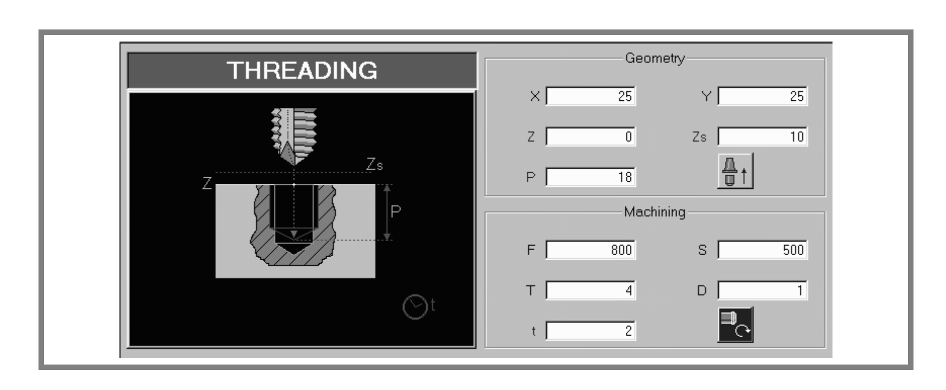
3. 快速 (G0) 运动到接近平面。
4. 以进给率 "F" 进刀 "I" 距离。
5. 钻孔循环直到到达总的深度 "P"。
   ·1· 快速返回 (G0) 替代距离 "B"。
   ·2· 快速接近 (G0) 直到距前次单步钻削到位置 1 mm 处。
   ·3· 以进给率 "F" 进刀 "I" 距离。
6. 暂停 "t" 秒。
7. 快速退回 (G0) 到安全平面 (Zs)。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

8. 快速运动 (G0) 到下一点 。
9. 钻削新孔，重复步骤 3, 4, 5, 6, 7。



12.5 攻螺纹



几何参数：

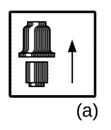
X, Y　加工点。

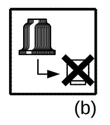
Z　工件表面坐标。

Zs　安全平面坐标。

P　总深度。

Kf　退出进给率因子。
刚性攻螺纹允许从螺纹孔快速退出，总是保持进给率和主轴速度之间的同步。退出进给率是进给率乘以因子 (Kf) 。


(a)


(b)

攻螺纹类型 (图标).

用离合器的攻螺纹 (a).

刚性攻螺纹 (b).

加工参数：

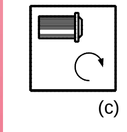
F　进给率。

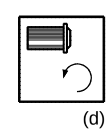
S　主轴速度。

T　刀具。

D　刀具偏置。

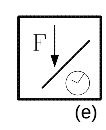
t　底面暂停，单位为秒。


(c)

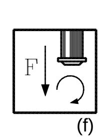

(d)

主轴旋转方向 (图标).

顺时针使用图标 (c) 和逆时针使用图标 (d)



(e)



(f)

进给率类型 (图标).

用单位 mm/min 或 (英寸 / 分钟)(e).

用单位毫米 / 转 (f).

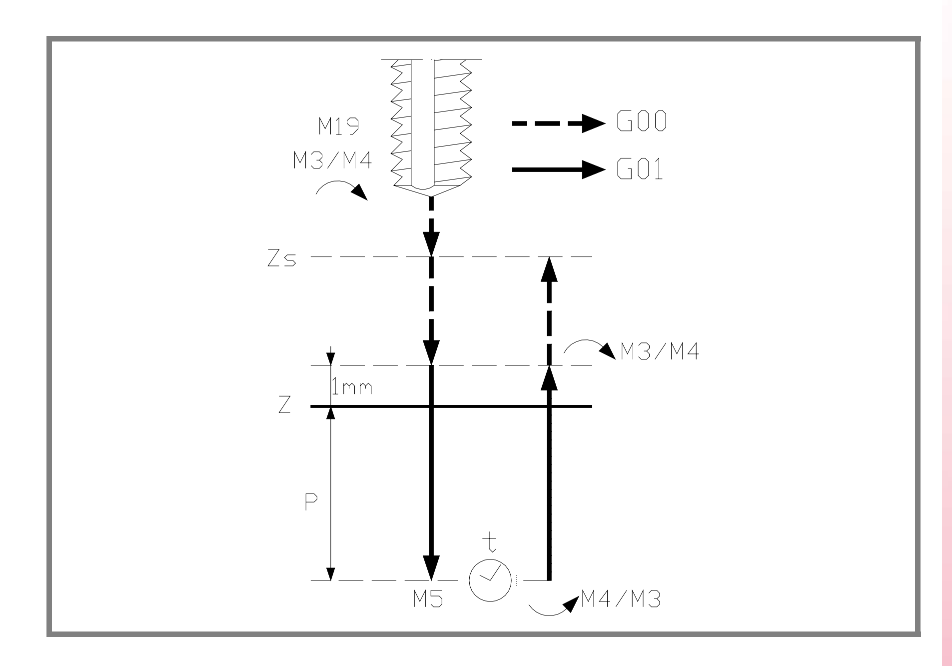


12.5.1 基本操作：

1. 如果是刚性攻螺纹，主轴定向 (M19)。

   如果是用离合器的攻螺纹方式，启动主轴以要求的方向旋转。
2. 快速运动 (G0), 直到 XY 点和安全平面 (Zs)。

   依据初始平面的位置，先在 XY 平面运动动，然后沿 Z 轴运动。反之亦然 。



3. 快速运动 (G0) 到接近平面 。
4. 攻螺纹。以 100% 的编写的进给率 "F" 和主轴速度 "S" 执行。用离合器的攻螺纹加工不能被中断 。刚性攻螺纹时，进给率倍率百分比可以被改变，甚至停止 (0% 倍率)。
5. 如果 "t" 不是 0, 主轴停止 (M05) 和暂停 。
6. 如果用离合器的攻螺纹，改变主轴旋转方向。
7. 退回，退出螺纹孔，到达接近平面。

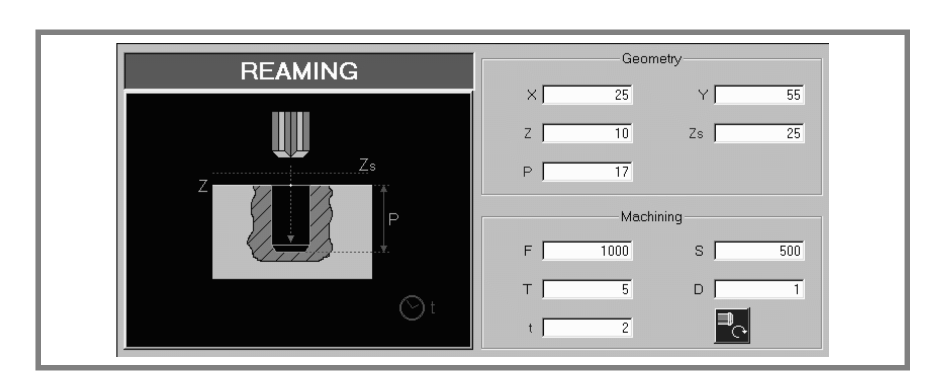
   以 100% 的编写的进给率 "F" 和主轴速度 "S" 。当用离合器的攻螺纹加工时，螺纹推出不能被中断。当刚性攻螺纹时，进给率倍率百分比可以被改变，甚至停止 (0% 倍率)。
8. 如果用离合器的攻螺纹加工时，改变主轴旋转方向 (恢复初始情况)。
9. 快速退回 (G0) 到安全平面 (Zs)。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

10. 快速运动 (G0) 到下一点 。
11. 钻削新孔，重复步骤 3, 4, 5, 6, 7, 8, 9。



12.6 铰孔



几何参数 :

X, Y 加工点。

Z 工件表面坐标。

Zs 安全平面坐标。

P 总加工深度。

加工参数 :

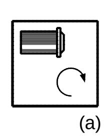
F 进给率。

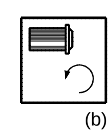
S 主轴速度。

T 刀具。

D 刀具偏置。

t 底面暂停，单位为秒


(a)


(b)

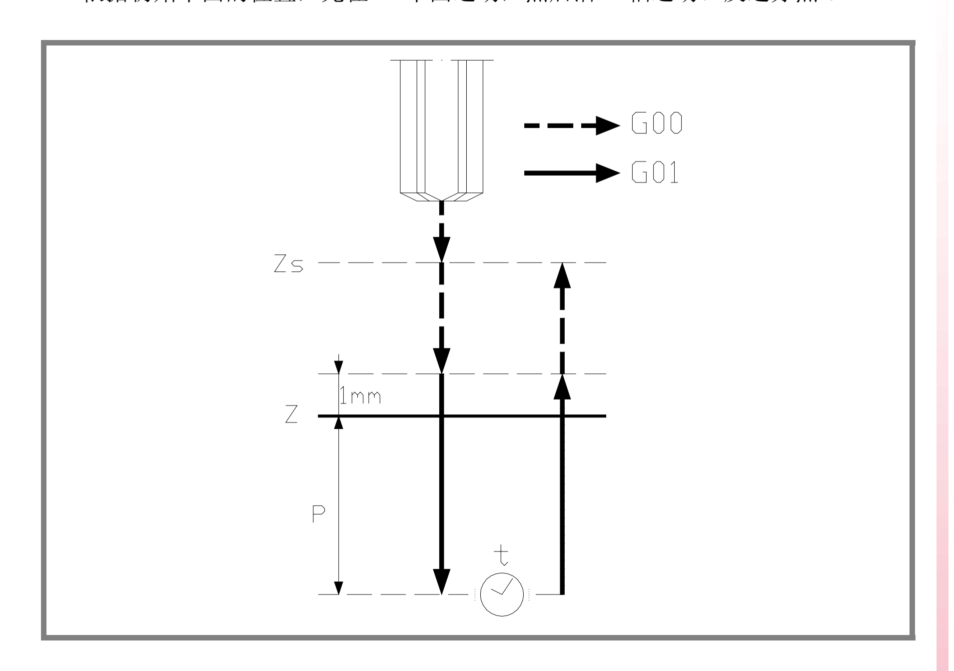
主轴旋转方向 (图标).

顺时针使用图标 (a) 和逆时针使用图标 (b)

12.6.1 基本操作：

1. 启动主轴以需要的方向旋转。
2. 快速运动 (G0), 直到 XY 点和安全平面 (Zs)。

   依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。



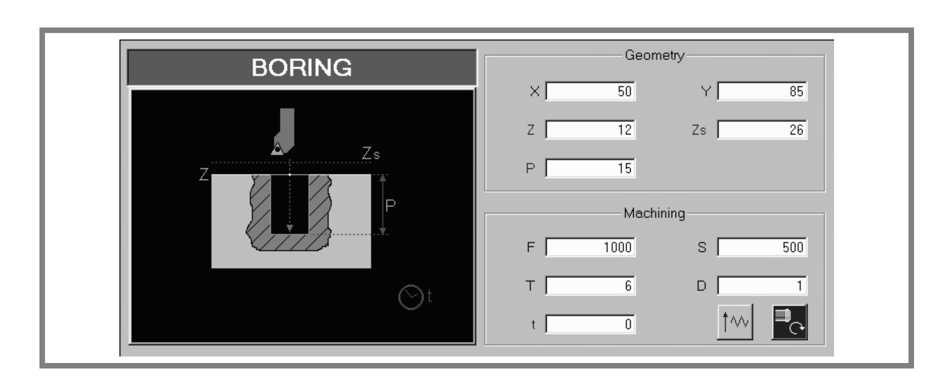
3. 快速运动 (G0) 到接近平面 。
4. 以进给率 "F" 进刀。
5. 暂停 "t" 秒。
6. 以进给率 "F" 退回到接近平面。
7. 快速运动 (G0) 到安全平面 (Zs)。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

8. 快速运动 (G0) 到下一点 。
9. 钻削新孔，重复步骤 3, 4, 5, 6, 7。



12.7 镗孔 1.



几何参数：

X, Y　加工点。

Z　工件表面坐标。

Zs　安全平面坐标。

P　总加工深度。

加工参数：

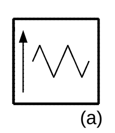
F　进给率。

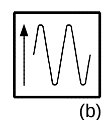
S　主轴速度。

T　刀具。

D　刀具偏置。

t　底面暂停，单位为秒

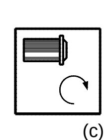

(a)

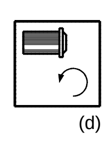

(b)

退回类型 (图标).

以进给率 "F" 方式，主轴旋转 。图标 (a).

以快速 (G0) 方式，主轴停转。图标 (b).


(c)


(d)

主轴旋转方向 (图标).

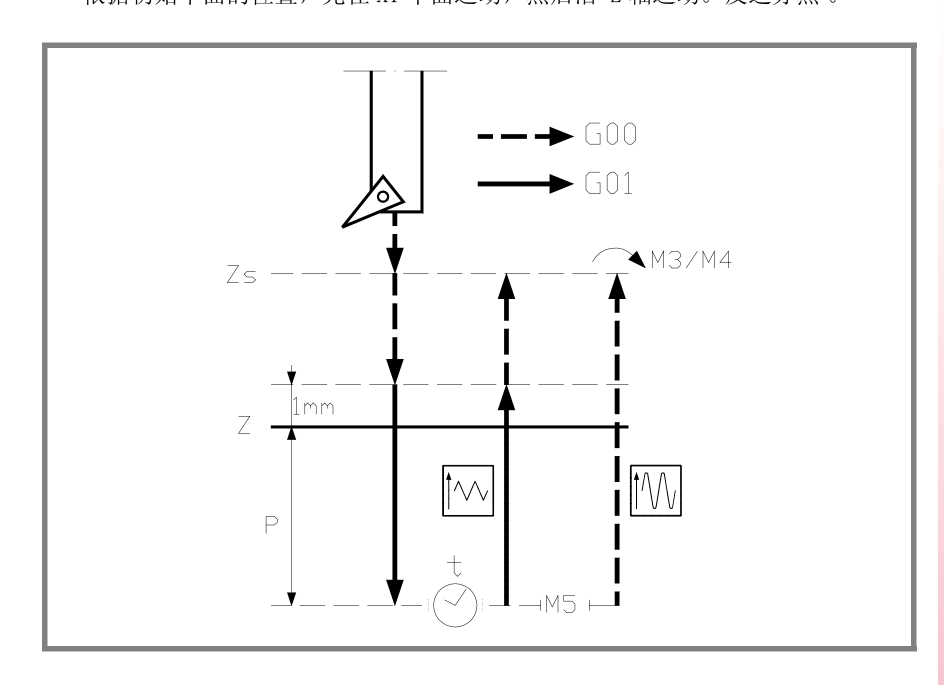
顺时针使用图标 (c) 和逆时针使用图标 (d)



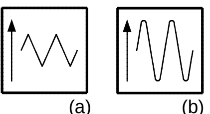
12.7.1 基本操作：

1. 启动主轴以需要的方向旋转。
2. 快速运动 (G0), 直到 XY 点和安全平面 (Zs)。

   依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。



3. 快速运动 (G0) 到接近平面 。
4. 以进给率 "F" 进刀。
5. 暂停 "t" 秒。
6. 如果图标 b) 被定义， 主轴停转 (M05)。
7. 退回。



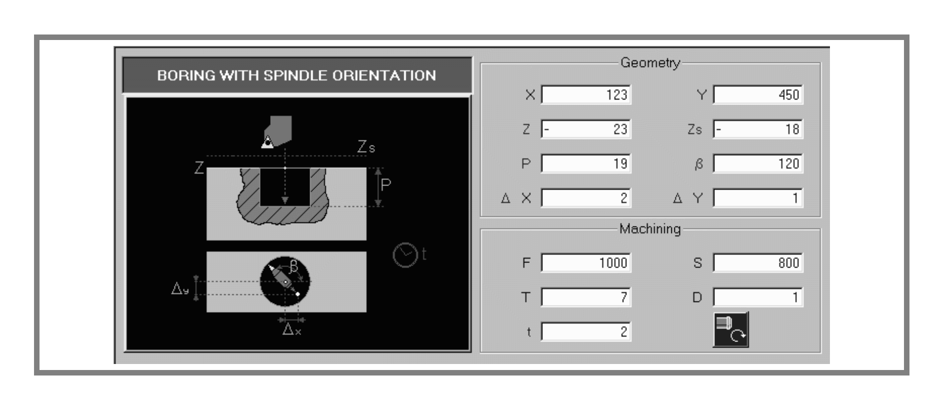
- 如果图标(a) 被定义，先以进给率"F"退回到接近平面(距工件上表面Z以上1mm)，然后快速 (G0) 运动到安全平面 Zs。
- 如果图标 (b) 被定义，以快速方式 (G0) 退回到安全平面，然后主轴以原来的旋转方向启转。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

8. 快速运动 (G0) 到下一点 。
9. 钻削新孔，重复步骤 3, 4, 5, 6, 7。



12.8 镗孔 2.



几何参数 :

X, Y　加工点。

Z　工件表面坐标。

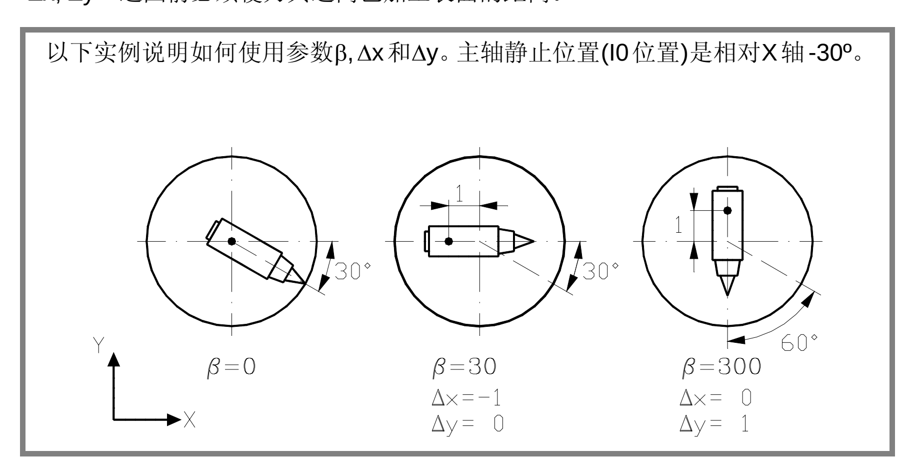
Zs　安全平面坐标。

P　总加工深度。

β　主轴定位角度，单位度。用于退回。

Δx, Δy　退回前必须使刀具远离已加工表面的距离。

以下实例说明如何使用参数β, Δx 和Δy。主轴静止位置(I0 位置)是相对X 轴 -30º。



加工参数 :

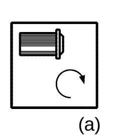
F　进给率。

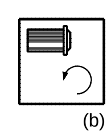
S　主轴速度。

T　刀具。

D　刀具偏置。

t　底面暂停，单位为秒。


(a)


(b)

主轴旋转方向 (图标).

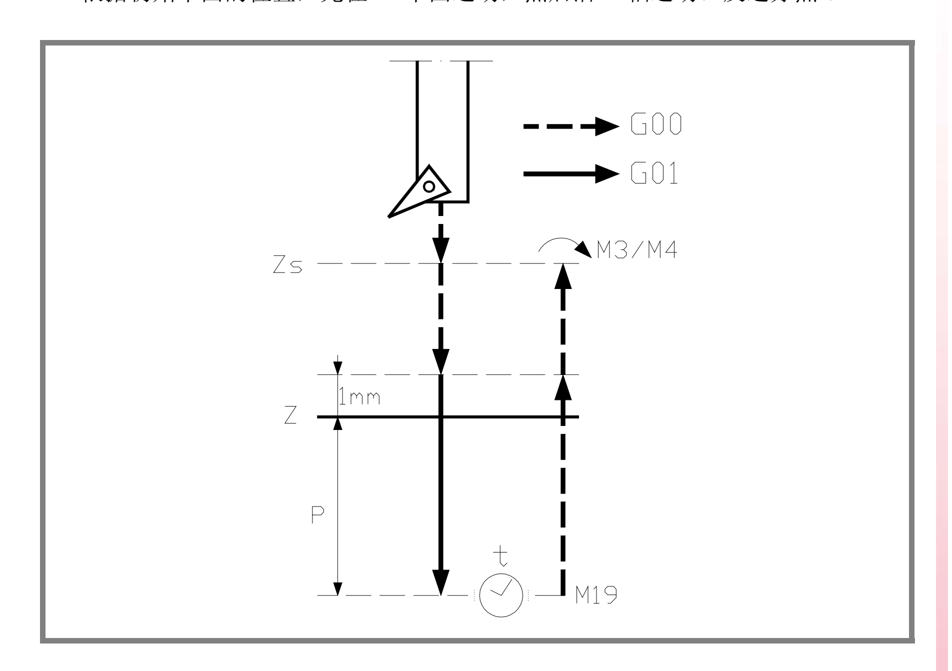
顺时针使用图标 (a) 和逆时针使用图标 (b)

12.8.1 基本操作：

1. 启动主轴以需要的方向旋转。

2. 快速运动 (G0), 直到 XY 点和安全平面 (Zs)。

依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。



3. 快速运动 (G0) 到接近平面 。

4. 以进给率 "F" 进刀。

5. 暂停 "t" 秒。

6. 主轴停止，刀具定位在 "β" 确定的角度位置 (M19)。

7. 使刀具离开已加工表面，移动 "Δx, Δy" 确定的距离。

8. 快速退回 (G0) 到接近平面。

9. 刀具返回 (XY) 位置，启动主轴以原来的方向旋转。

10. 快速运动 (G0) 到安全平面 (Zs)。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

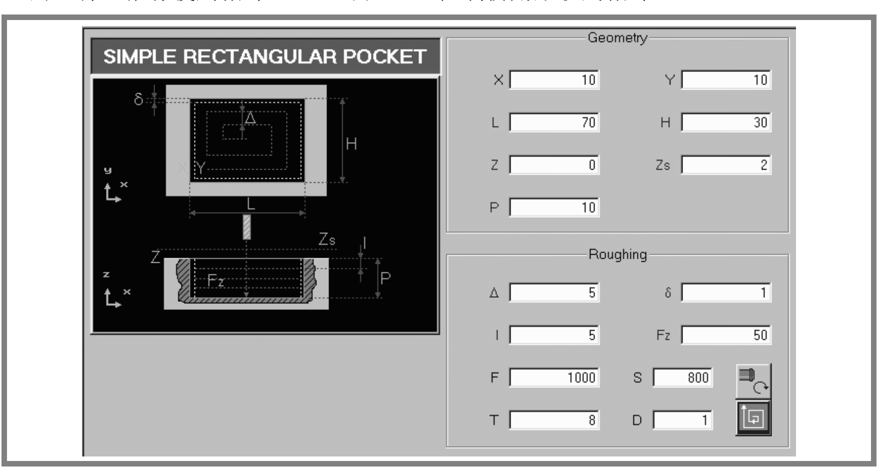
11. 快速运动 (G0) 到下一点 。

12. 钻削新孔，重复步骤 3, 4, 5, 6, 7,8, 9, 10.。



12.9 简单型腔加工

循环的粗加工和精加工程序段结合的方法是先前由用户用指令 #HSC, G5, G50 或设置的一种。推荐使用指令 #HSC 或 G5， 控制拐角形状用指令 #ROUNDPAR。

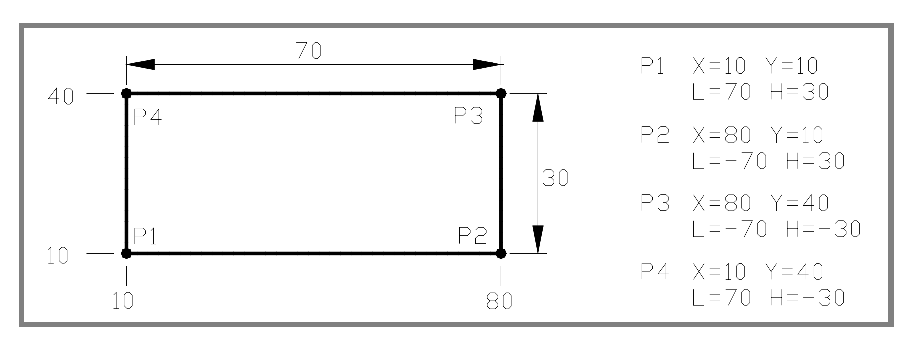


几何参数 :

X, Y　　型腔角坐标。

L, H　　型腔尺寸。

符号表示相对 XY 点的方向。



Z　　工件表面坐标。

Zs　　安全平面坐标。

P　　总加工深度。

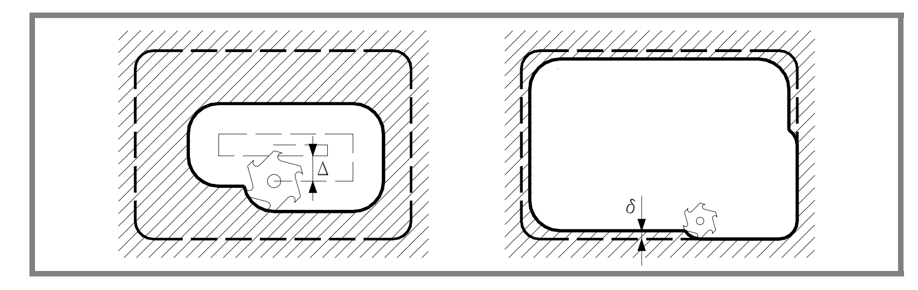
加工参数 :

Δ　　最大铣削宽度

循环重新计算加工宽度，所以所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。



δ　　侧壁面的精加工余量。

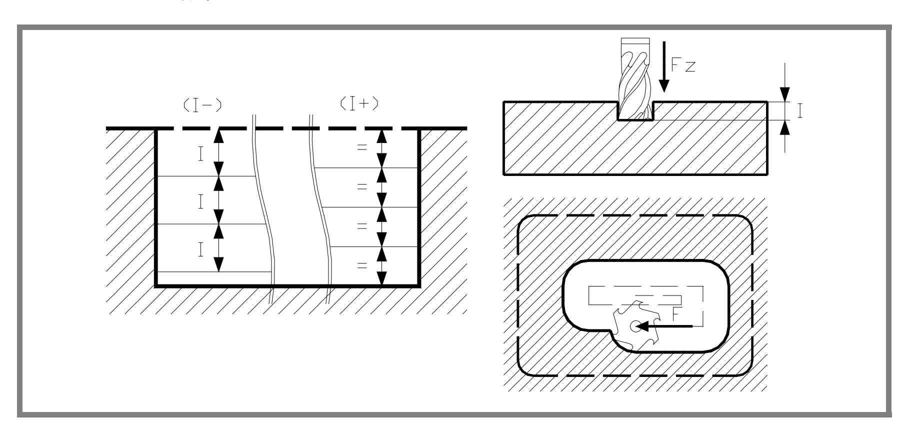


I　　每步进刀量

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号 (I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况下，循环限制单步进给量小于刀具表中所赋的切削长度值。

Fz　　进刀进给率

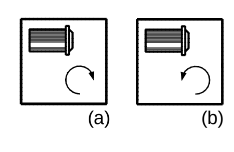


F　　表面铣削进给率。

S　　主轴速度。

T　　刀具。

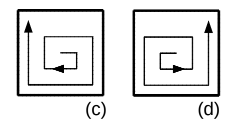
D　　刀具偏置。



主轴旋转方向 (图标)

顺时针使用图标 (a)

逆时针使用图标 (b)



加工方向 (图标)

顺时针使用图标 (c)

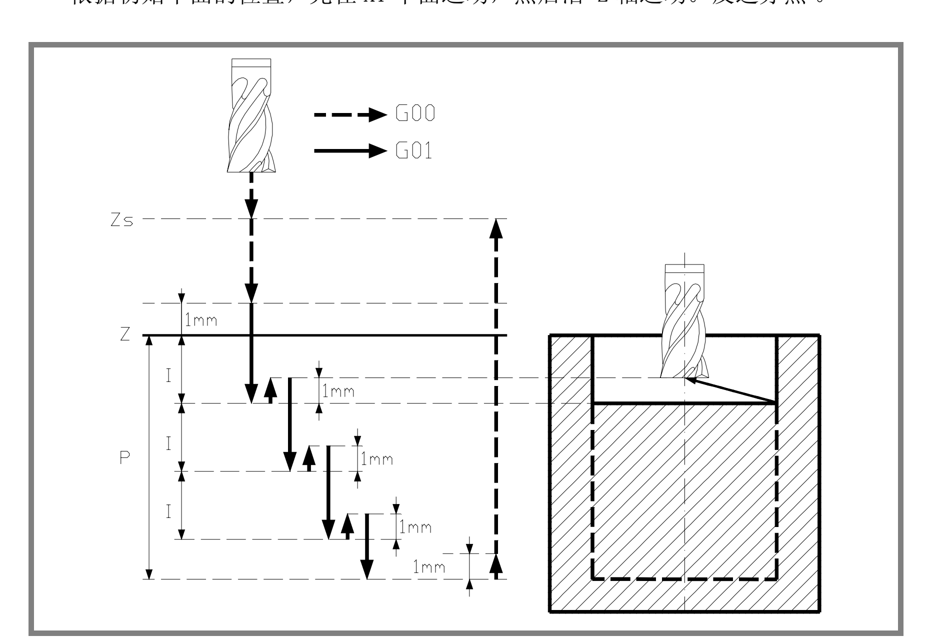
逆时针使用图标 (d)



12.9.1 基本操作：

1. 启动主轴以需要的方向旋转。
2. 快速运动 (G0) 到型腔中心和安全平面 (Zs) 的位置。

   依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。



3. 快速运动 (G0) 到接近平面 。
4. 首次进刀，进给率 "Fz" ， 进刀量 "I"。
5. 型腔表面铣削。

   粗加工以进给率 "F" 值 ， 加工最大宽度 "Δ" 值，精加工余量 "δ" 值进行加工。

   精加工以切向进刀和退出，进给率 "F"，切削量 "δ" 进行加工。
6. 快速退回 (G0) 到接近平面上型腔的中心位置 。
7. 新的平面铣削，一直到满足型腔总的加工深度要求。

   ·1· 进刀，以 "Fz" 确定的进给率，从先前已加工到的表面起进刀 "I" 确定的值。

   ·2· 铣削新的型腔表面，同 5 和 6 说明的步骤。
8. 快速退回 (G0) 到安全平面 (Zs)。

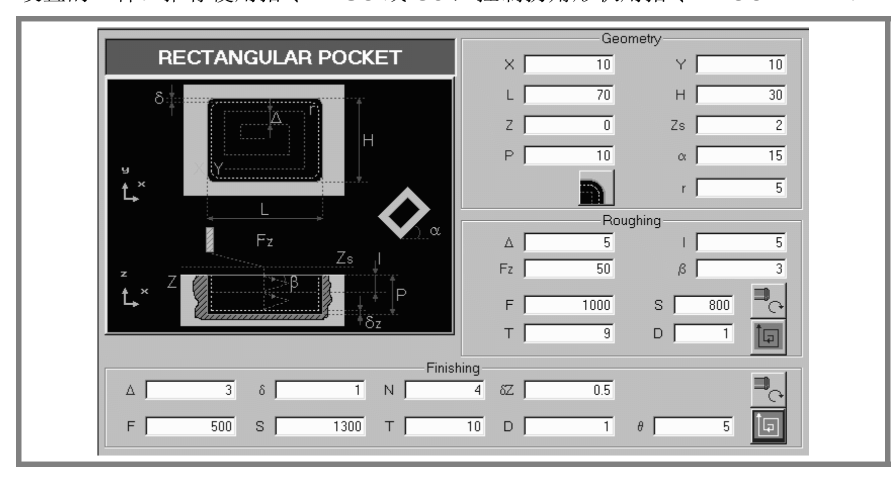
如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

9. 快速运动 (G0) 到下一点 。
10. 钻削新孔，重复步骤 3, 4, 5, 6, 7,8。



12.10 矩形型腔加工

循环的粗加工和精加工程序段被结合的方法是先前由用户用指令 #HSC, G5, G50 或设置的一种。推荐使用指令 #HSC 或 G5 ，控制拐角形状用指令 #ROUNDPAR。

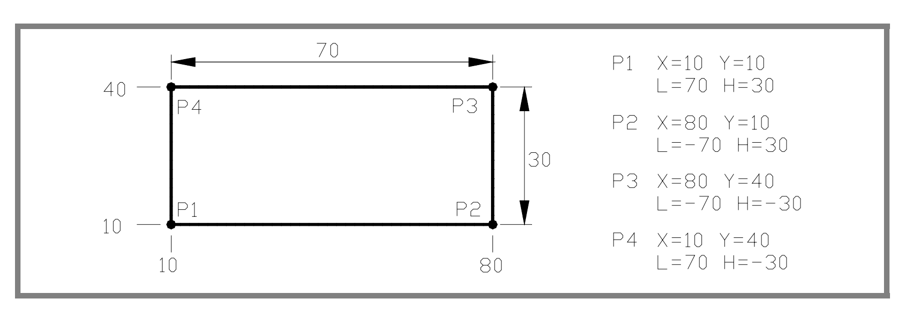


几何参数 :

X, Y　型腔角

L, H　型腔尺寸

符号表示相对 XY 点的方向。

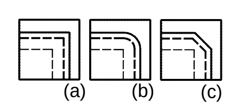


Z　工件表面坐标

Zs　安全平面坐标

P　总加工深度

το　型腔和横坐标的夹角，单位度。以定义的型腔角 X,Y 点来执行旋转。



拐角类型 (图标).

方角使用图标 (a).

圆角使用图标 (b).

倒角使用图标 (c).

r　圆角半径或倒角尺寸。



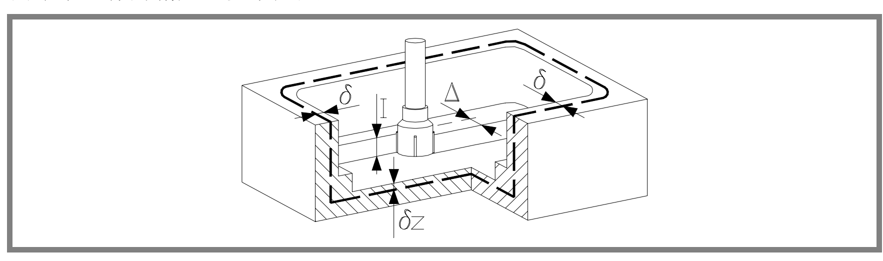
粗加工参数：

粗加工操作把型腔掏空，只留下列精加工余量：

δ　　型腔侧壁精加工余量。

δz　　型腔底面精加工余量。

两个余量作为精加工参数定义。



粗加工操作定义参数如下：

Δ　　最大铣削宽度。

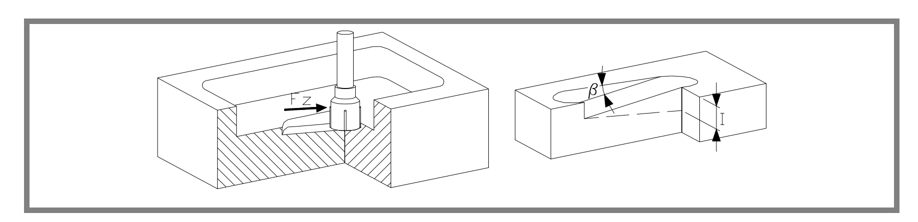
循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

I　　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号 (I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制单步进给量小于刀具表中所赋的切削长度值。

Fz　　进刀进给率



β　　进刀角度。

在型腔的中心开始和结束时，以 Z 字形执行进刀。
如果定义了一个大于在刀具表中赋给刀具的值，采用刀具表中的值。

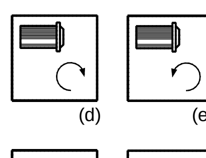
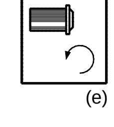
F　　表面铣削进给率。

S　　主轴速度。

T　　粗加工刀具。

如果编写 T=0, 没有粗加工。

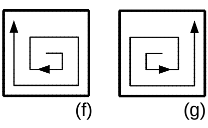


主轴旋转方向 (图标)。

顺时针使用图标 (d)。

逆时针使用图标 (e)。



加工方向 (图标).

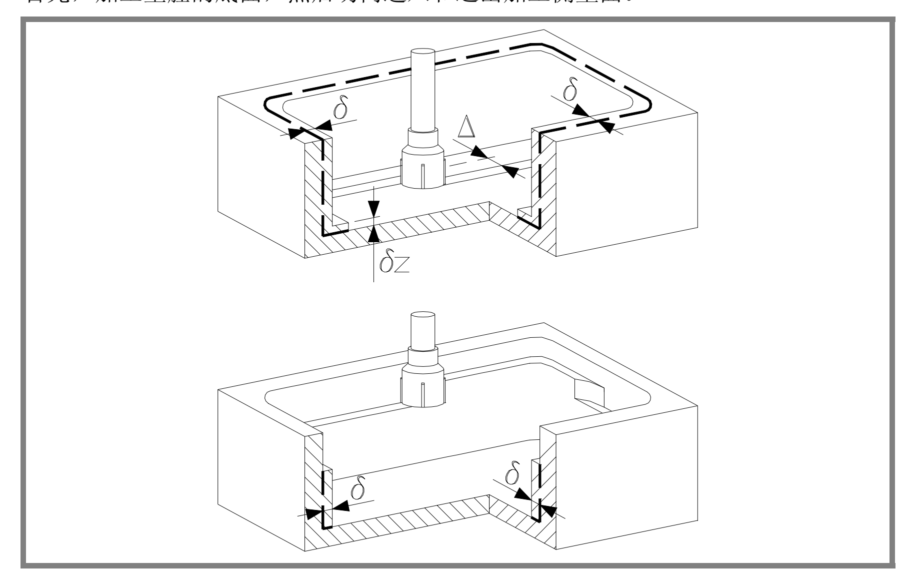
顺时针使用图标 (f).

逆时针使用图标 (g).

精加工参数 :

精加工操作分两个阶段执行。

首先，加工型腔的底面，然后切向进入和退出加工侧壁面。



精加工操作定义参数如下：

δ 侧壁面的精加工余量。

δz 型腔底面的精加工余量。

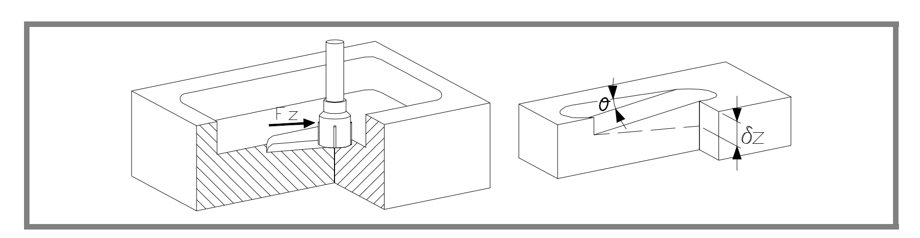
Δ 型腔底面的铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

N 侧面精加工的进刀的数量（步数）。如果作为结果的一步大于刀具表中赋予刀具的切削长度 ，步长受表中值的限制。

θ 进刀角度。

进刀以粗加工参数设置的参数 "Fz" 进给率执行在型腔中心的开始和结束。T
如果定义的值大于刀具表中赋予刀具的值，采用表中的值。





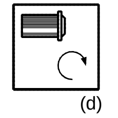
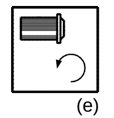
F 表面和侧面的铣削进给率。

S 主轴进给率。

T 精加工刀具。

如果编写 T=0，没有精加工。

D 刀具偏置。

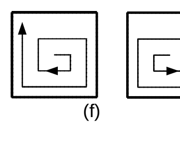



主轴旋转方向 (图标)。

顺时针使用图标 (d)。

逆时针使用图标 (e)。

加工方向 (图标).



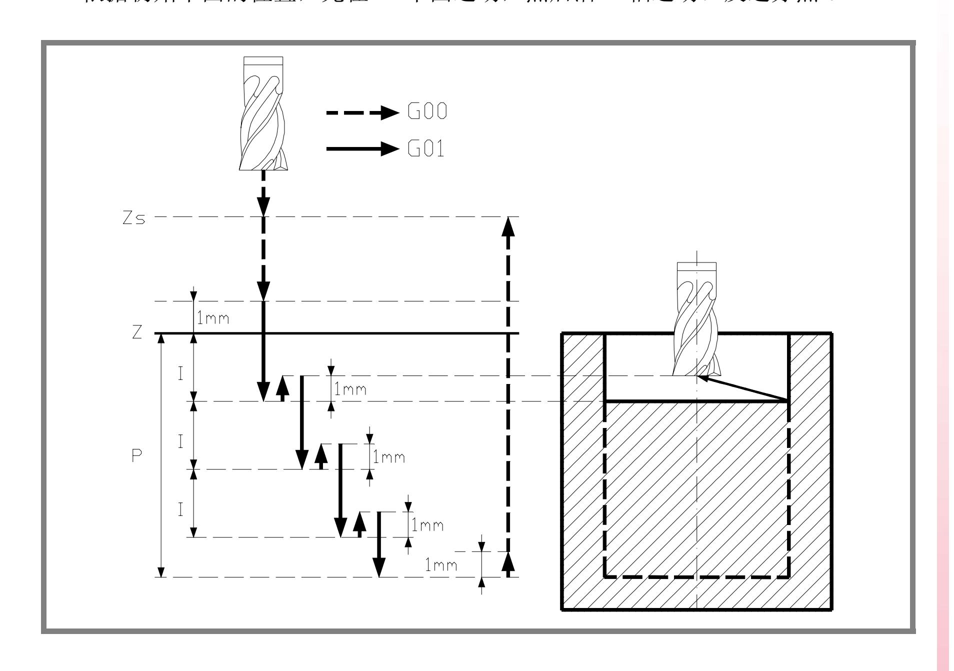
顺时针使用图标 (f)。

逆时针使用图标 (g)。

12.10.1 基本操作 :

1. 选择粗加工刀具，启动主轴以需要的方向旋转。
2. 快速运动 (G0) 到安全平面 (Zs) 在型腔中心的位置。

   依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。



3. 快速运动 (G0) 到接近平面 。
4. 粗加工操作。

   以层加工，直到到达总加工深度减去底面的精加工余量 "δz"。

   ·1· 以角度 "β"，进给率 "Fz" 进刀 "I" 。

   ·2· 铣削型腔表面直到距型腔壁"δ"的距离。以进给率"F"执行，如果需要，重新计算最大切削宽度 (Δ) ，这样所有的切削宽度一致。

   ·3· 快速退回 (G0) 到型腔中心，距已加工表面 1 mm 处。
5. 快速退回 (G0) 一直到安全平面 (Zs)。
6. 选择精加工刀具，快速 (G0) 接近粗加工成型底面上 1 mm 处。
7. 精加工型腔底面。

   ·1· 以角度 "θ"，进给率 "Fz" 进刀。

   ·2· 铣削型腔底面直到距型腔壁"δ"的距离。以精加工进给率"F"执行，如果需要，重新计算精加工最大铣削宽度 (Δ) ，这样所有的切削宽度一致。
8. 快速 (G0) 退回到接近平面 (距型腔表面 "Z" 上 1 mm 处) 上的型腔中心。
9. 精加工侧壁面。

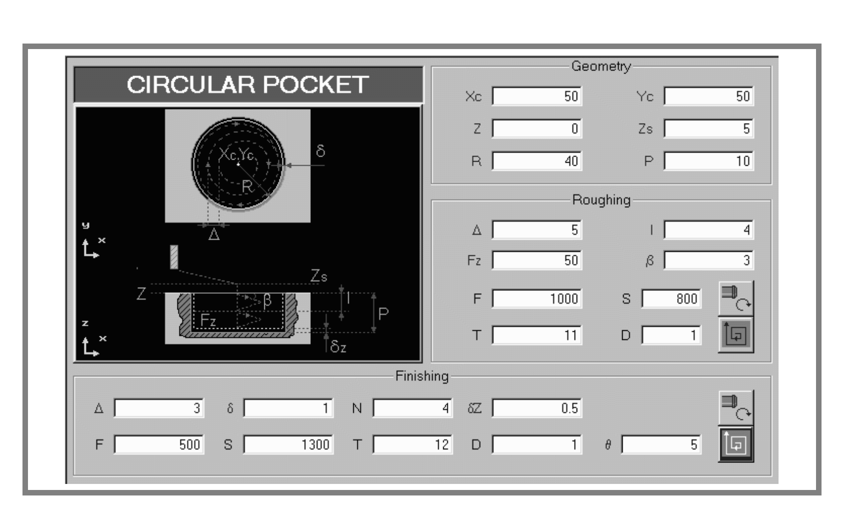
   以进给率 "F" ，切入和退出，执行 "N" 次进刀。
10. 快速退回 (G0) 安全平面 (Zs) 上的型腔中心 。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤:

11. 快速运动 (G0) 到下一点 。
12. 钻削新孔，重复步骤 3, 4, 5, 6, 7,8, 9, 10。



12.11 圆形型腔加工



几何参数 :

Xc, Yc 型腔中心。

R 型腔半径。

Z 工件表面坐标。

Zs 安全平面坐标。

P 总加工深度。

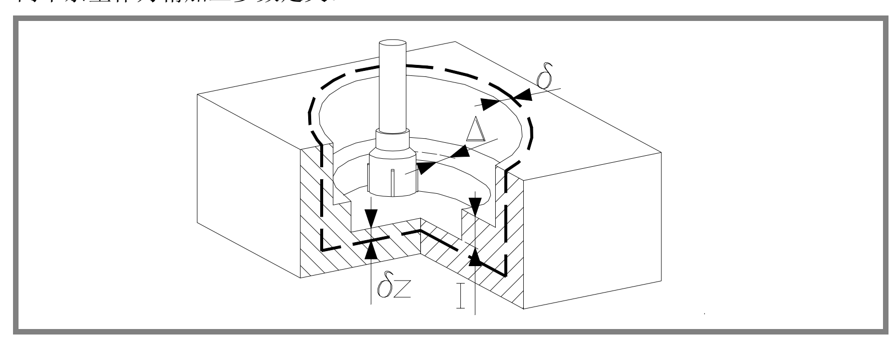
粗加工参数 :

粗加工操作把型腔加工空，只留下列精加工余量:

δ 型腔侧壁面的精加工余量。

δz 型腔底面的精加工余量。

两个余量作为精加工参数定义。



粗加工操作定义参数如下:

Δ 最大铣削宽度

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

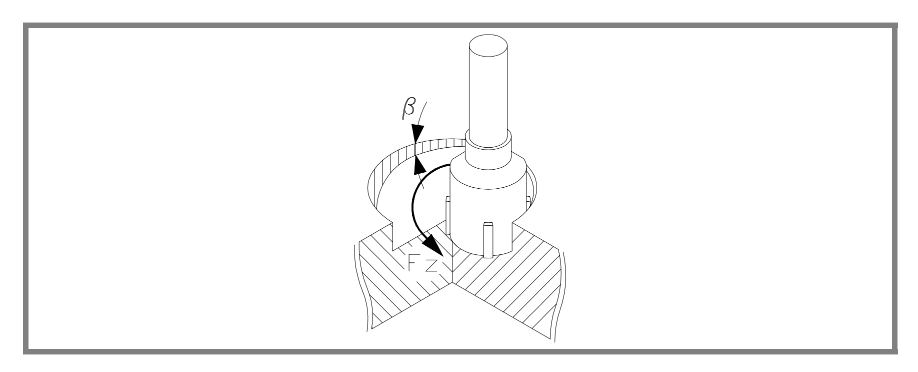


I 每步进刀量

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号 (I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况， 循环限制单步进给量小于刀具表中所赋的切削长度值。

Fz 进刀进给率。



β 进刀角度。

在型腔的中心开始和结束，以螺旋线执行进刀。
如果定义了一个大于在刀具表中赋给刀具的值，采用刀具表中的值。

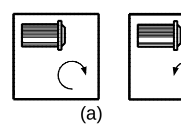
F 表面铣削进给率。

S 主轴速度。

T 粗加工刀具。

如果编写 T=0, 没有粗加工 。

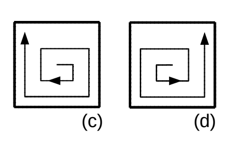
D 刀具偏置。



主轴旋转方向 (图标)。

顺时针使用图标 *(a)*。

逆时针使用图标 *(b)*。



加工方向 (图标)。

顺时针使用图标 *(c)*。

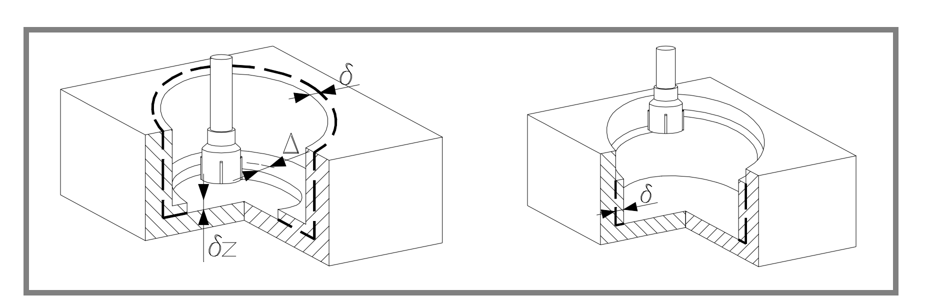
逆时针使用图标 *(d)*。



精加工参数：

精加工操作分两个阶段执行。

首先，加工型腔的底面，然后切向进入和退出加工侧壁面。



精加工操作定义参数如下：

δ　　型腔壁面精加工余量。

δz　　型腔底面精加工余量。

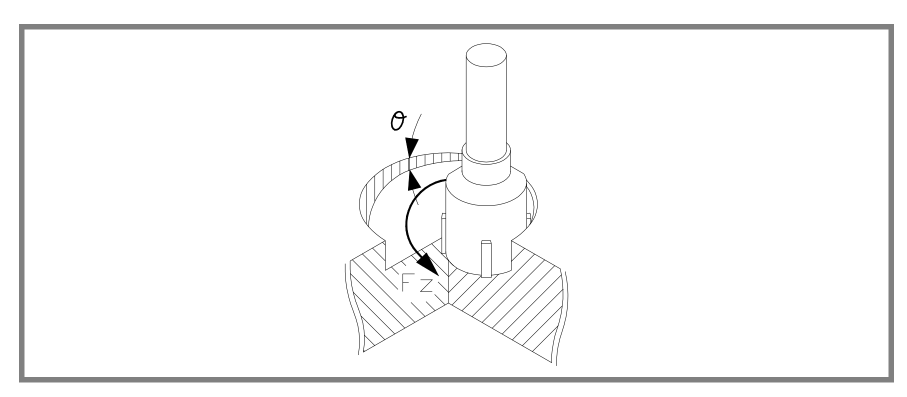
Δ　　型腔底面铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

N　　侧面精加工的进刀的数量（步数）。如果作为结果的一步大于刀具表中赋予刀具的切削长度 ，步长受表中值的限制。

θ　　进刀角度。

进刀沿螺旋线路径，以粗加工参数设置的参数 "Fz" 进给率执行在型腔中心的开始和结束。 T
如果定义的值大于刀具表中赋予刀具的值，采用表中的值。



F　　表面和侧壁面铣削进给率。

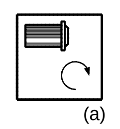
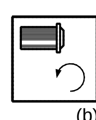
S　　主轴速度。

T　　精加工刀具。

如果编写 T=0, 没有精加工。

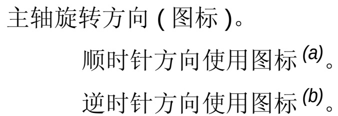
D　　刀具偏置。



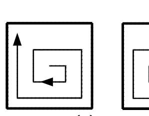
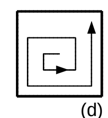
(a) (b)

主轴旋转方向 (图标)。



顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。

(c) (d)

加工方向 (图标)。

顺时针方向使用图标 (c)。

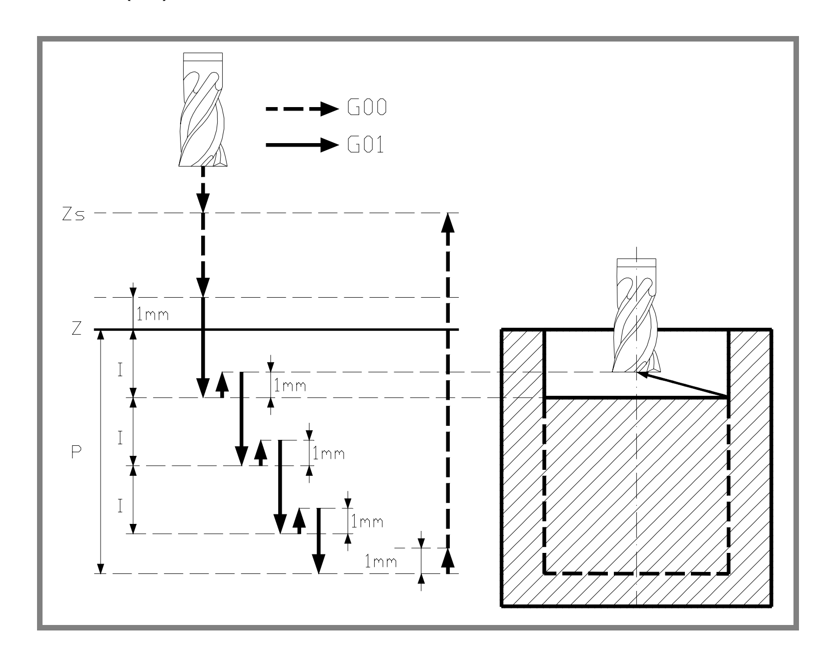
逆时针方向使用图标 (d)。



12.11.1 基本操作：

1. 选择粗加工刀具，启动主轴以需要的方向旋转。
2. 快速运动 (G0) 到安全平面 (Zs) 在型腔中心的位置。

   依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然。
3. 快速运动 (G0) 直到接近平面 。



4. 粗加工操作。

   以层加工，直到到达总加工深度减去底面的精加工余量 "δz"。

   ·1· 以角度 "β"，进给率 "Fz" 进刀 "I" 。

   ·2· 铣削型腔表面直到距型腔壁"δ"的距离。以进给率"F"执行，如果需要，重新计算最大切削宽度 (Δ) ，这样所有的切削宽度一致。

   ·3· 快速退回 (G0) 到型腔中心，距已加工表面 1 mm 处。
5. 快速退回 (G0) 直到安全平面 (Zs)。
6. 选择精加刀具，快速接近已粗加工底面上 1 mm 处。
7. 精加工型腔底面。

   ·1· 以角度 "θ"，进给率 "Fz" 进刀。

   ·2· 铣削型腔底面直到距型腔壁"δ"的距离。以精加工进给率"F"执行，如果需要，重新计算精加工最大铣削宽度 (Δ) ，这样所有的切削宽度一致。
8. 快速退回 (G0) 到接近表面上型腔中心处 。
9. 精加工侧壁面。

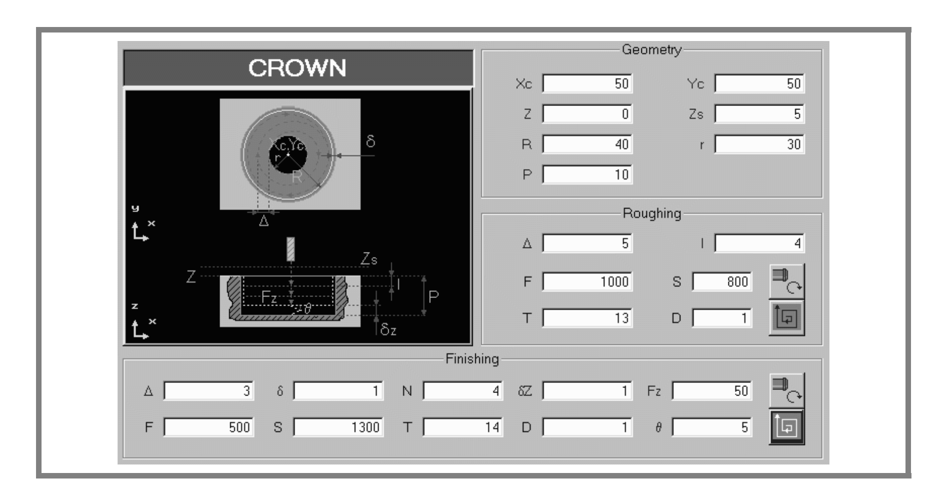
   以进给率 "F" ，切入和退出，执行 "N" 次进刀。
10. 快速退回 (G0) 到安全平面上 (Zs) 型腔中心处 。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

11. 快速运动 (G0) 到下一点 。
12. 钻削新孔，重复步骤 3, 4, 5, 6, 7,8, 9, 10。



12.12 预制空腔型腔加工



几何参数：

Xc, Yc　型腔中心。

R　　型腔半径。

r　　先已加工空型腔半径。

Z　　工件表面坐标。

Zs　　安全平面坐标。

P　　总加工深度。

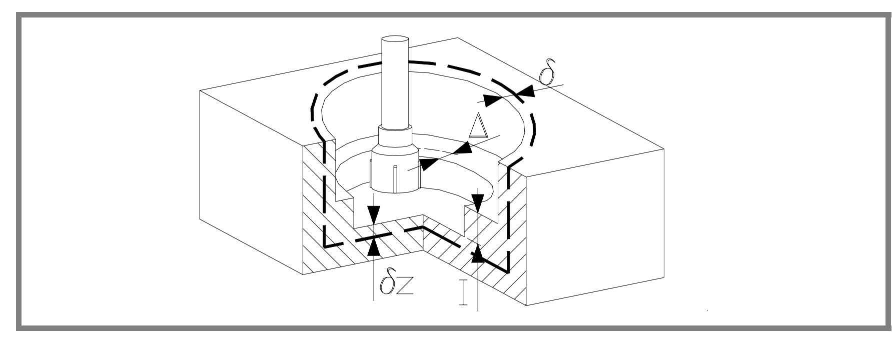
粗加工参数：

粗加工操作把型腔加工空，只留下列精加工余量：

δ　　型腔侧壁面的精加工余量。

δz　　型腔底面的精加工余量。

两个加工余量作为精加工参数定义。



粗加工操作定义参数如下：

Δ　　最大铣削宽度

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。

如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

I 每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号(I-),型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制单步进给量小于刀具表中所赋的切削长度值。

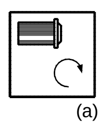
F 表面铣削进给率。

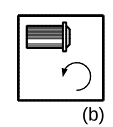
S 主轴速度。

T 粗加工刀具。

如果编写 T=0，没有粗加工。

D 刀具偏置。

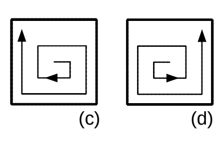

(a)


(b)

主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。


(c) (d)

加工方向 (图标)。

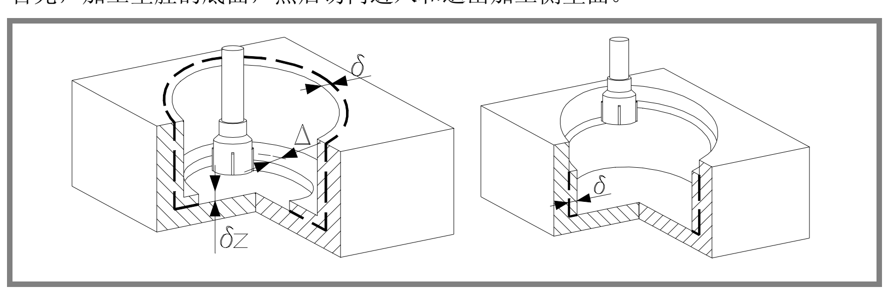
顺时针方向使用图标 (c)。

逆时针方向使用图标 (d)。

精加工参数：

精加工操作分两个阶段执行。

首先，加工型腔的底面，然后切向进入和退出加工侧壁面。



精加工操作定义参数如下：

δ 型腔侧壁面的精加工余量。

δz 型腔底面的精加工余量。

Δ 型腔底面的铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

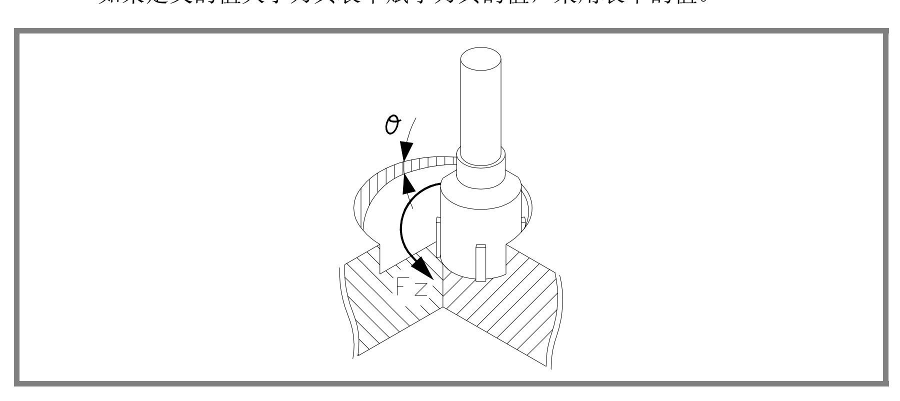


N　　侧面精加工的进刀的数量（步数）。如果作为结果的一步大于刀具表中赋予刀具的切削长度 ，步长受表中值的限制。

Fz　　进刀进给率。

θ　　进刀角度。

进刀沿螺旋线路径，以粗加工参数设置的参数 "Fz" 进给率执行在型腔中心的开始和结束。
如果定义的值大于刀具表中赋予刀具的值，采用表中的值。



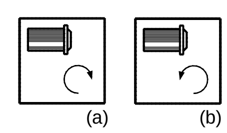
F　　表面和侧壁面铣削进给率。

S　　主轴速度。

T　　精加工刀具。

如果编写 T=0, 没有精加工。

D　　刀具偏置。

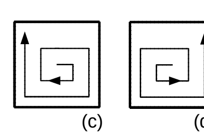


主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。

加工方向 (图标)。



顺时针方向使用图标 (c)。

逆时针方向使用图标 (d)。

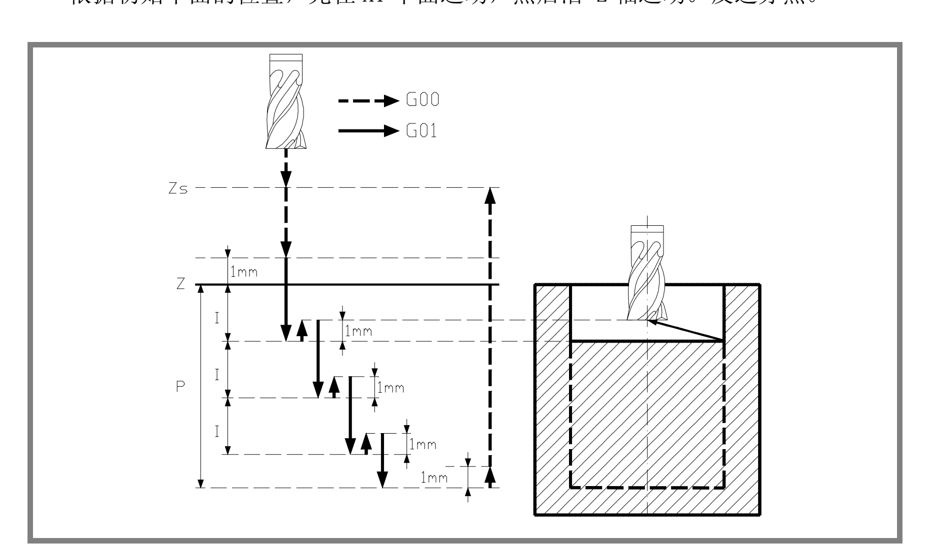


12.12.1 基本操作：

1. 选择粗加工刀具，启动主轴以需要的方向旋转。

2. 快速运动 (G0) 到安全平面 (Zs) 在型腔中心的位置。

依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然。



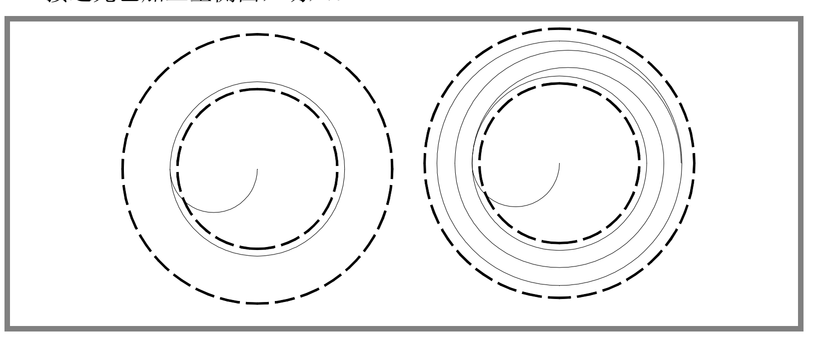
3. 快速运动 (G0) 直到接近平面。

4. 粗加工操作。

以层执行加工，直到到达总加工深度减去底面的精加工余量 "δz"。

·1· 进刀 "I" 值。

·2· 接近先已加工空侧面，切入。



·3· 铣削型腔表面直到距型腔壁"δ"的距离。以进给率"F"执行，如果需要，重新计算最大切削宽度 (Δ) ，这样所有的切削宽度一致。

·4· 快速退回 (G0) 到型腔中心，距已加工表面 1 mm 处。

5. 快速退回 (G0) 直到安全平面 (Zs).

6. 选择精加工刀具，快速接近已粗加工底面上 1 mm 处。

7. 精加工型腔底面。

·1· 以角度 "θ"，进给率 "Fz" 进刀。

·2· 铣削型腔底面直到距型腔壁"δ"的距离。以精加工进给率"F"执行，如果需要，重新计算精加工最大铣削宽度 (Δ) ，这样所有的切削宽度一致。

8. 快速退回 (G0) 到接近表面 (距工件表面 "Z" 上 1 mm 处) 上型腔中心处 。



9. 精加工侧壁面。

以进给率 "F"， 切入和退出，执行 "N" 次进刀。

10.快速退回 (G0) 到安全表面 (Zs) 上型腔中心处 。

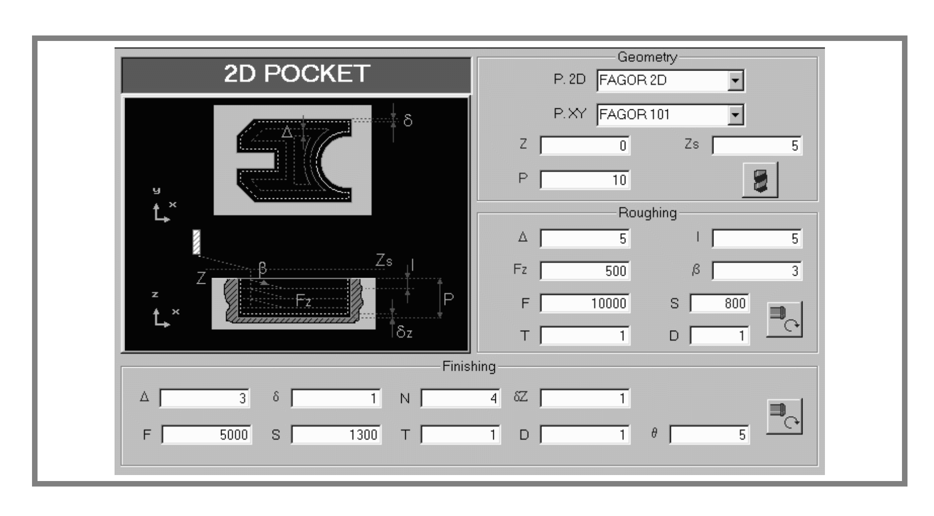
如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

11.快速运动 (G0) 到下一点 。

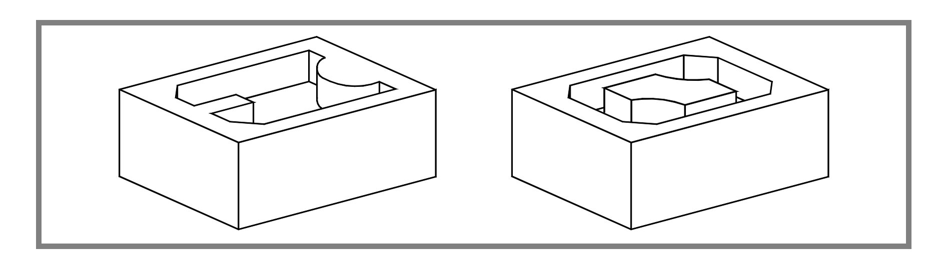
12.钻削新孔，重复步骤 3, 4, 5, 6, 7,8, 9, 10。



12.13 2D 型腔加工



这种型腔由外轮廓和一定数量叫做岛的内轮廓组成。所有的 2D 型腔壁面是垂直的 。



推荐先定义指令 #ROUNDPAR，为了获得好的精加工效果，因为精加工过程以 G05 执行。

几何参数 :

型腔组成和平面轮廓存储在 \ Cnc8070\ Users\ Profile。

pocket.P2D　　型腔组成。

profile.PXY　　平面轮廓。

P.2D　2D 型腔名称。

一旦型腔外形有效，数控系统使型腔几何图形和名称联合。

P.XY　平面轮廓名称。

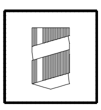
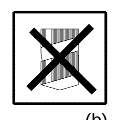
轮廓必须确定型腔的外轮廓和那些岛。

Z　工件表面坐标。

Zs　安全平面坐标。

P　总加工深度。





(a) (b)

钻孔 (图标).

说明在加工型腔之前进行钻孔 (a) 或还是不进行钻孔 (b). 用于当粗加工刀具不能向下加工时。

按下 "Drilling" 软键访问钻削循环，定义后，按下 "End" 软键返回 2D 型腔循环。

钻削刀具的尺寸必须不超过粗加工刀具的半径：或如果没有粗加工操作，用于粗加工底面。

循环依据编程轮廓和粗加工刀具计算钻孔点。

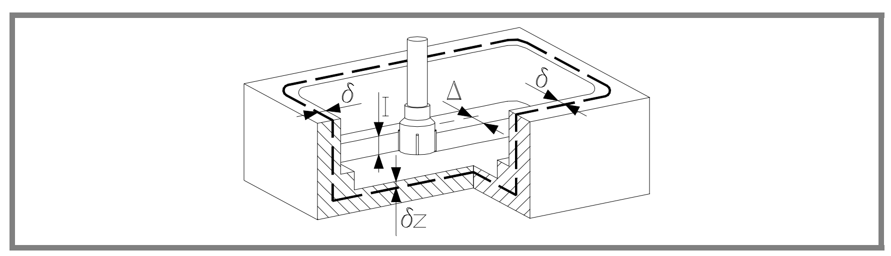
粗加工参数 :

粗加工操作把型腔加工空，只留下列精加工余量：:

δ 型腔侧壁面精加工余量。

δz 型腔底面精加工余量。

两个余量作为精加工参数定义。



粗加工操作定义参数如下 :

Δ 最大铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4。

I 每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号 (I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

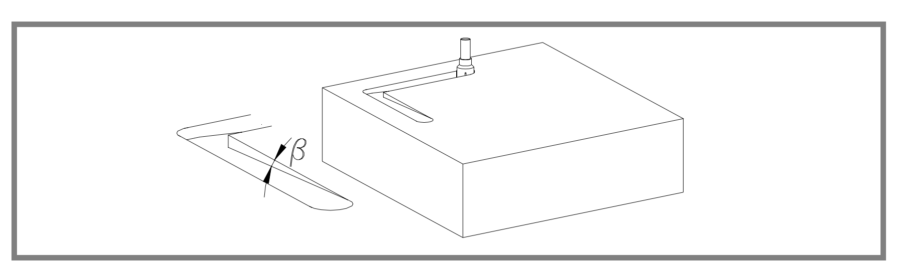
在任何一种情况，循环限制单步进给量小于刀具表中所赋的切削长度值。

Fz 进刀进给率。



β　　进刀角度。

执行进刀时保持这个角度直到到达相应的深度。
如果定义了一个大于在刀具表中赋予刀具的值，采用刀具表中的值。

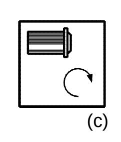


F　　表面铣削进给率。

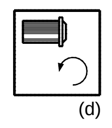
S　　主轴速度。

T　　粗加工刀具。

如果编写 T=0，没有粗加工 。



(c)



(d)

主轴旋转方向 (图标)。

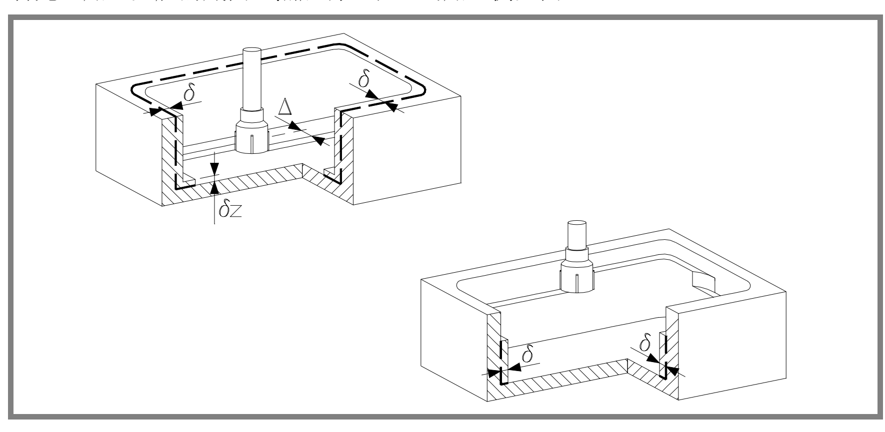
顺时针方向使用图标 (c)。

逆时针方向使用图标 (d)。

精加工参数 :

精加工操作分两个阶段执行。

首先，加工型腔的底面，然后切入和退出加工侧壁面。



精加工操作定义参数如下：

δ　　型腔侧壁面精加工余量。

δz　　型腔底面精加工余量。

Δ　　型腔底面铣削宽度。

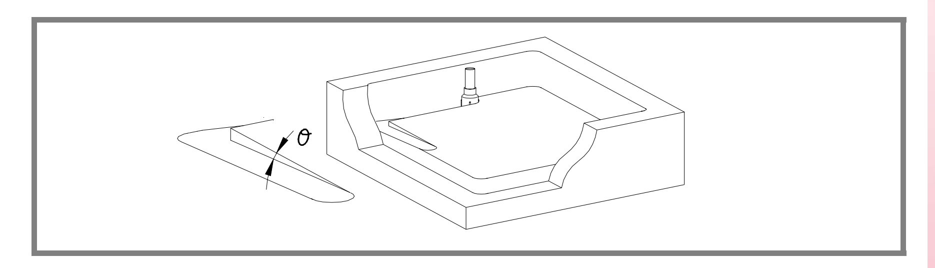
循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

N　　侧面精加工的进刀的数量 （步数）。

当编写 0 值，根据刀具表中赋予刀具的刀具长度值，执行可能的最少的进刀数量。

θ　　进刀角度。

进刀以粗加工参数设置的参数 "Fz" 进给率执行，保持该角度直到到达相应的深度。
如果定义的值大于刀具表中赋予刀具的值，采用表中的值。



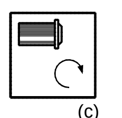
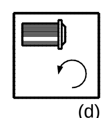
F　　表面和侧壁面铣削进给率。

S　　主轴速度。

T　　精加工刀具。

如果编写 T=0，没有精加工。

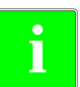
D　　刀具偏置。


(c)

(d)

主轴旋转方向 (图标)。
顺时针方向使用图标 (c)。
逆时针方向使用图标 (d)。

12.13.1 可执行型腔加工文件

模拟或执行此类型的型腔加工，数控系统使用一个有几何信息的可执行文件。该文件在执行或模拟型腔加工的第一时间生成。如果从编辑器修改某些型腔几何数据或是使用的刀具，数控系统会再次生成此文件。



在 V2.00 以前的版本里，在嵌入循环之前，用户从编辑器生成可执行文件。
从 V2.00 版本开始，当需要时数控系统负责生成可执行文件。

可执行文件存储在目录 CNC8070 \Users \Pocket 下，用型腔的名称 (参数 P.2D) 和扩展名 *C2D*。这些文件不可以删除、移动到其它位置或是以任何方式损坏。如果当执行或是模拟型腔加工时，数控系统不能找到这些文件，将会生成这些文件。

一般的， 2D 型腔由以下文件组成 。

pocket.P2D	型腔组成。
profile.PXY	平面轮廓。
pocket.C2D	可执行文件。

当执行或模拟型腔加工时，在软件升级后，可执行文件同时升级。

12.13.2 基本操作

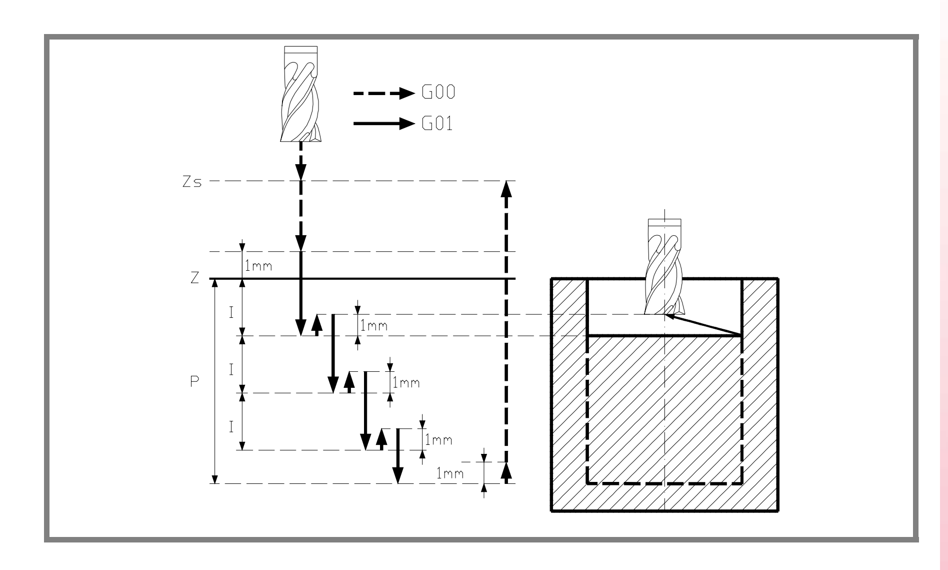
数控系统依据型腔的几何信息和刀具半径计算初始坐标。

1. 钻削操作。如果编写了此步骤。

2. 选择粗加工刀具，启动主轴以需要的方向旋转。

3. 快速运动 (G0) 到安全平面 (Zs) 在型腔中心的位置。

依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然。



4. 快速运动 (G0) 直到接近平面。

5. 粗加工操作。

以层加工，直到到达总加工深度减去底面的精加工余量 "δz"。

·1· 以角度 "β"，进给率 "Fz" 进刀 "I" 。

·2· 铣削型腔表面直到距型腔壁"δ"的距离。以进给率"F"执行，如果需要，重新计算最大切削宽度 (Δ) ，这样所有的切削宽度一致。

执行以后的路径是和轮廓同心圆的，以外轮廓被定义的一样的方向加工。

岛以相反的方向进行加工。

·3· 快速退回 (G0) 到型腔中心，距已加工表面 1 mm 处。

6. 快速退回 (G0) 到安全平面 (Zs)。

7. 选择精加工刀具，快速 (G0) 接近距粗加工完成的地面上方 1 mm 处。

8. 精加工型腔底面。

·1· 以角度 "θ"，进给率 "Fz" 进刀。

·2· 铣削型腔底面直到距型腔壁"δ"的距离。以精加工进给率"F"执行，如果需要，重新计算精加工最大铣削宽度 (Δ) ，这样所有的切削宽度一致。

执行以后的路径是和轮廓同心圆的，以外轮廓定义的一样的方向加工。

岛以相反的方向进行加工。

9. 快速 (G0) 退回到接近平面。



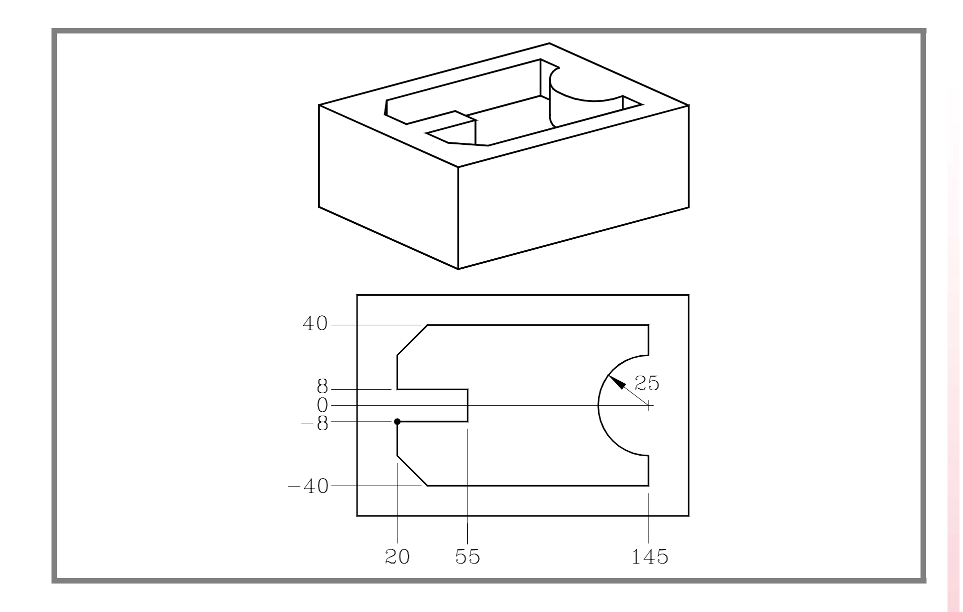
10.精加工侧壁面。

以进给率 "F" ，切入和退出，执行 "N" 次进刀。

外轮廓同被定义的方向加工，岛以反方向加工。

11.快速退回 (G0) 到安全表面 (Zs) 。

12.13.3 如何定义 2D 轮廓的实例



轮廓 P.XY	FAGOR 101	[调用]

设置 :

横坐标轴 : X	纵坐标轴 : Y
自动缩放 : Yes	确认

轮廓 :

起点	X 20	Y -8		确认
直线	X 20	Y -40		确认
直线	X 145	Y -40		确认
直线	X 145	Y -25		确认
顺时针圆弧	Xf 145	Yf 25	R 25	确认
直线	X 145	Y 40		确认
直线	X 20	Y 40		确认
直线	X 20	Y 8		确认
直线	X 55	Y 8		确认
直线	X 55	Y -8		确认
直线	X 20	Y -8		确认

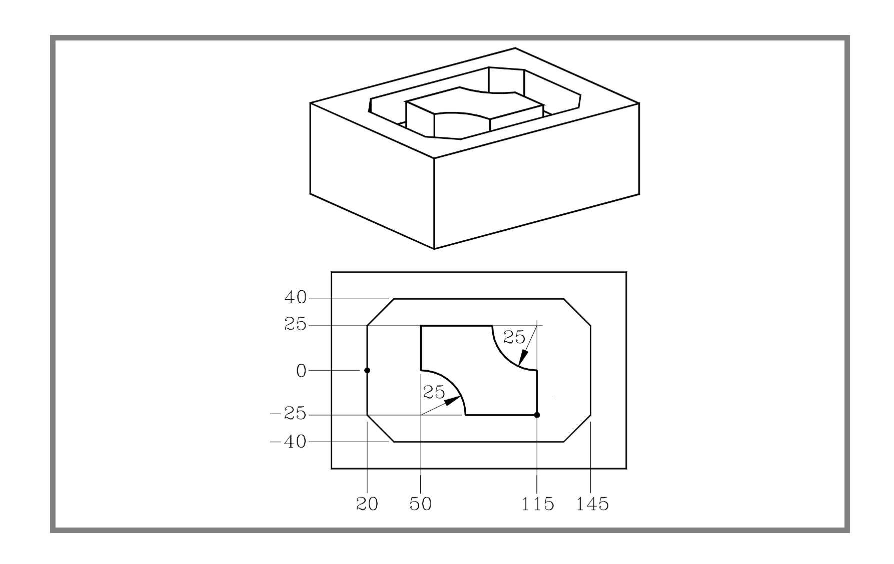
拐角

倒角	
选择左下拐角	[回车]
倒角 15	[回车]
选择左上拐角	[回车]
倒角 15	[回车]
	[ESC]

结束 :

保存轮廓





Profile P.XY	FAGOR 102	[调用]

设置 :

横坐标轴 : X	纵坐标轴 : Y
自动缩放 : Yes	确认

轮廓 (外轮廓):

起点	X 20	Y 0	确认
直线	X 20	Y -40	确认
直线	X 145	Y -40	确认
直线	X 145	Y 40	确认
直线	X 20	Y 40	确认
直线	X 20	Y 0	确认

拐角

倒角	
选择左下拐角	[回车]
倒角 15	[回车]
选择右下拐角	[回车]
倒角 15	[回车]
选择右上拐角	[回车]
倒角 15	[回车]
选择左上拐角	[回车]
倒角 15	[回车]
	[ESC]



新轮廓 (岛):

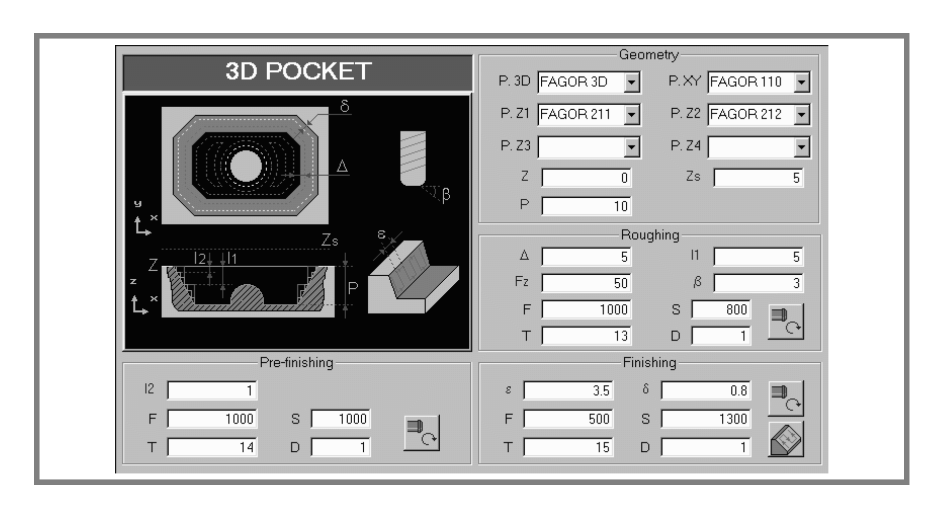
起点	X 115	Y -25		确认
直线	X 115	Y 0		确认
顺时针圆弧	Xf 90	Yf 25		
	Xc 115	Yc 25	R 25	确认
直线	X 50	Y 25		确认
直线	X 50	Y 0		确认
顺时针圆弧	Xf 75	Yf -25		
	Xc 50	Yc -25	R 25	确认
直线	X 115	Y -25		确认

结束 :

保存轮廓



12.14 3D 型腔加工

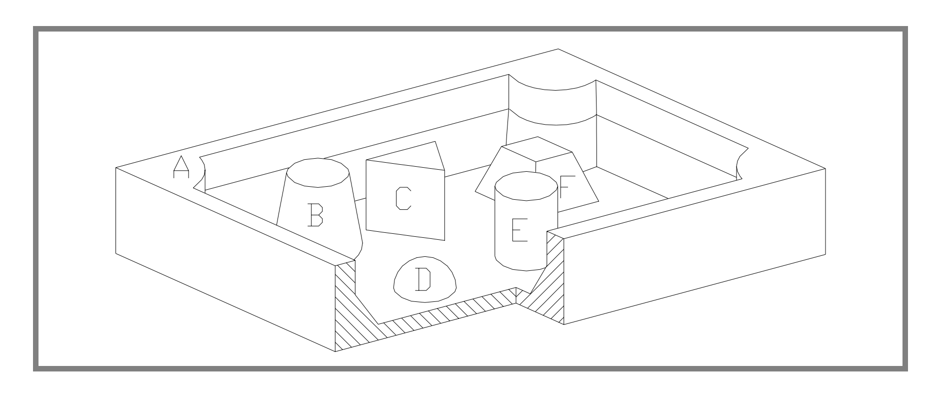


型腔由一个外轮廓和许多的叫做岛的内轮廓组成。

不同于 2D 型腔 , 它的壁面是垂直的 , 3D 型腔可以被定义为对于每个轮廓 有深度轮廓 (最多 4 个不同的).

表面轮廓定义所有轮廓，外轮廓和内轮廓 （岛）。

表面轮廓定义的前4 个轮廓可以被赋予它们自己的深度轮廓。其余的轮廓是垂直的。



该 3D 型腔轮廓有两个 “垂直轮廓”轮廓 (C 和 E)，四个 “非垂直轮廓”轮廓 (A, B, D 和 F)。

因为仅四个轮廓可以定义为 “非垂直轮廓”，所以必须先定义 A, B, D, F ，最后定义 C, E。

推荐预先定义定义指令 #ROUNDPAR，为了得到好的粗糙度，因为精加工路径以 G05 方式加工。



几何参数：

型腔的组成和平面、深度轮廓存储在 \ Cnc8070\ Users\ Profile。

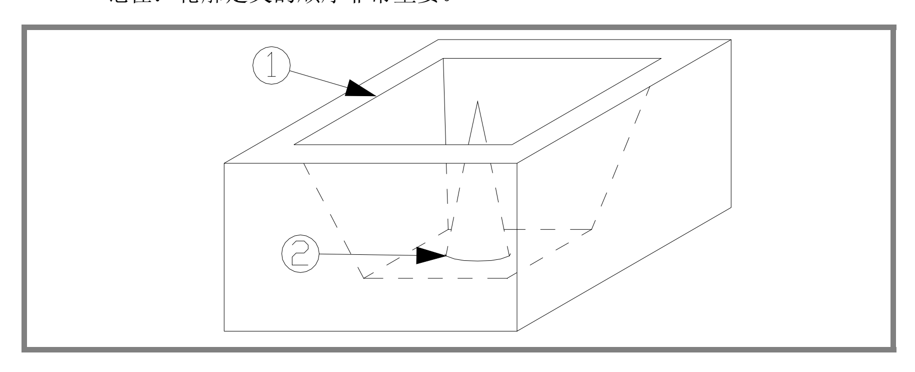
pocket.P3D	型腔组成。
profile.PXY	平面轮廓。
profile.PXZ	深度轮廓。

P.3D 3D 型腔的名字

一旦型腔配置确定了，数控系统使型腔几何图形联合到它的名字 (表面轮廓和深度轮廓)。

P.XY 表面轮廓或平面轮廓的名字。

必须确定所有轮廓。
对于外轮廓，文件对于表面 (1)。
对于岛，文件对于底部 (2)。
多有轮廓必须是闭合的，不能相互交叉。
记住：轮廓定义的顺序非常重要。



P.Z1 P.Z2 P.Z3 P.Z4

深度轮廓名字

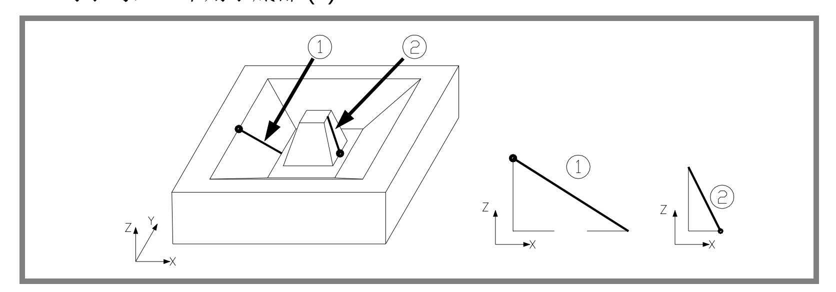
它们对应在表面轮廓定义的前四个轮廓，名字表示顺序。

为定义深度轮廓，使用平面的一根轴和垂直轴。

用相同的点定义轮廓的开始和深度轮廓的开始。

对于外轮廓，一个用于表面 (1).

对于岛，一个用于底部 (2).

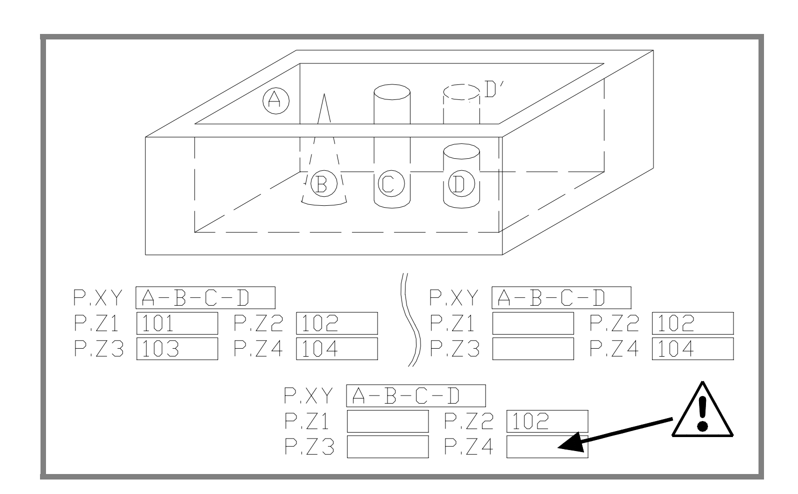


所有轮廓必须是开放的，没有沿它们路径的方向变化 (没有之字形)。

延伸出表平面的，外轮廓和内轮廓的垂直深度轮廓不需要编写。



图示三个编程实例。



当在表面轮廓定义轮廓时，所有情形下遵循顺序 A-B-C-D。

上侧左手实例定义所有深度轮廓 : Z1(A), Z2(B), Z3(C), Z4(D)。

上侧左手实例省略所有垂直深度轮廓 : Z1(A), Z3(C)。

下侧的实例编写错误，因为垂直轮廓一个也没有定义。

如果岛 (D) 的轮廓没有定义，循环认为是到延伸到表平面，将会加工岛 (D').。

Z 工件表面坐标。

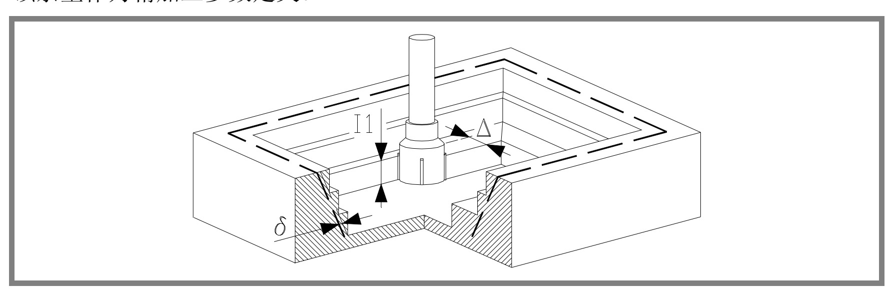
Zs 安全平面坐标。

P 总加工深度。

粗加工参数 :

粗加工操作加工空型腔，侧壁面留精加工余量 δ :

该余量作为精加工参数定义。



粗加工参数定义如下 :

Δ 最大铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4。



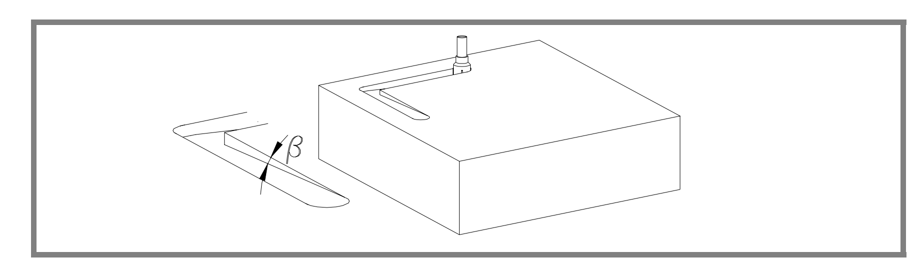
I1　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号 (I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制单步进给量小于刀具表中所赋的切削长度值。

Fz　进刀进给率。

β　进刀角度。
执行进刀时保持这个角度直到到达相应的深度。
如果定义了一个大于在刀具表中赋予刀具的值，采用刀具表中的值。



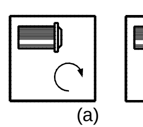
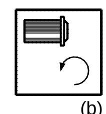
F　表面铣削进给率。

S　主轴速度。

T　粗加工刀具。

如果编写 T=0, 没有粗加工。

D　刀具偏置。

(a)　(b)

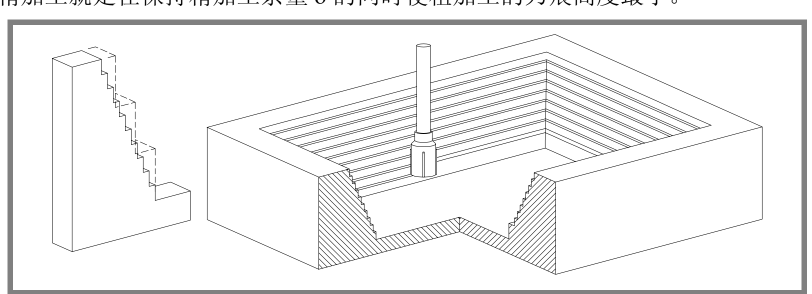
主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。

半精加工参数 :

半精加工就是在保持精加工余量 δ 的同时使粗加工的刀痕高度最小。



半精加工定义参数如下 :

I2　　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号(I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制单步进给量小于刀具表中所赋的切削长度值。

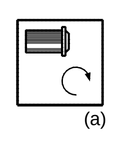
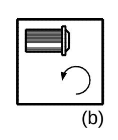
F　　铣削进给率。

S　　主轴速度。

T　　半精加工刀具。

如果编写 T=0, 没有半精加工。

D　　刀具偏置。


(a)

(b)

主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

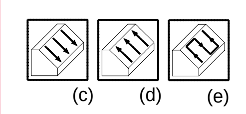
逆时针方向使用图标 (b)。

精加工参数 :

精加工操作考虑刀尖几何尺寸，补偿刀具表中定义的刀尖半径。

δ　　型腔侧壁面精加工余量。

ε　　侧壁面铣削宽度。


(c)　(d)　(e)

侧壁面加工方向 (图标).

一直向下 (c), 一直向上 (d), Z 字形 (e)。

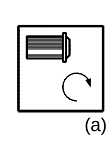
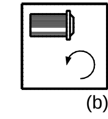
F　　铣削进给率。

S　　主轴速度。

T　　精加工刀具。

如果编写 T=0, 没有精加工。

D　　刀具偏置。


(a)

(b)

主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。

12.14.1 可执行型腔加工文件

模拟或执行这种类型的型腔，数控系统使用一个有几何信息的可执行文件。该文件在执行或模拟型腔加工的第一时间生成。如果从编辑器修改某些型腔几何数据或是使用的刀具，数控系统会再次生成此文件。

i

在 V2.00 以前的版本里，在嵌入循环之前，用户从编辑器生成可执行文件。
从 V2.00 版本开始，当需要时数控系统负责生成可执行文件。

可执行文件存储在目录 CNC8070 \Users \Pocket 下，用型腔的名称 (参数 P.3D) 和扩展名 *C3D*。这些文件不可以删除、移动到其它位置或是以任何方式损坏。如果当执行或是模拟型腔加工时，数控系统不能找到这些文件，将会生成这些文件。

一般地 , 3D 型腔由下列文件组成。

pocket.P3D	型腔组成。
profile.PXY	平面轮廓。
profile.PXZ	深度轮廓。
pocket.C3D	可执行文件。

当执行或模拟型腔加工时，在软件升级后，可执行文件同时升级。



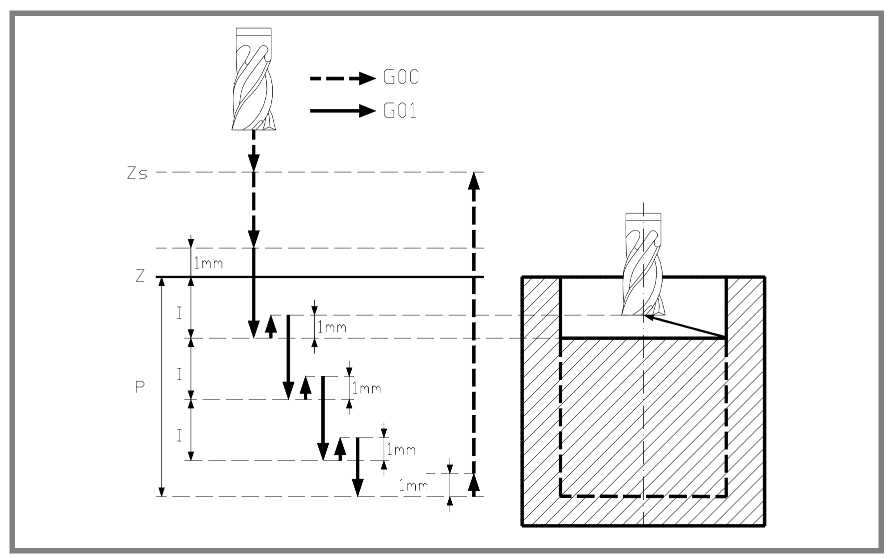
12.14.2 基本操作

数控系统依据型腔的几何信息和刀具半径计算初始坐标。

1. 选择粗加工刀具，启动主轴以需要的方向旋转。

2. 快速运动 (G0) 到安全平面 (Zs) 在型腔中心的位置。

依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然。



3. 快速运动 (G0) 到接近平面。

4. 粗加工操作。

以层加工，直到到达总加工深度。

·1· 以角度 "β"，进给率 "Fz" 进刀 "I1"。

·2· 铣削型腔表面直到距型腔壁"δ"的距离。以进给率"F"执行，如果需要，重新计算最大切削宽度 (Δ)，这样所有的切削宽度一致。

执行以后的路径是和轮廓同心圆的，以外轮廓被定义的一样的方向加工。

岛以相反的方向被加工。

·3· 快速退回 (G0) 到型腔中心，距已加工表面 1 mm 处。

5. 快速退回 (G0) 到接近平面 。

6. 选择半精加工刀具，启动主轴以需要的方向旋转。

7. 半精加工型腔侧壁面。

以 "F" 半精加工进给率，"I2" 确定的进刀量执行。

外轮廓加工同定义的方向，岛的加工方向相反。

8. 快速退回 (G0) 到接近平面 。

9. 选择精加工刀具，启动主轴以需要的方向旋转。

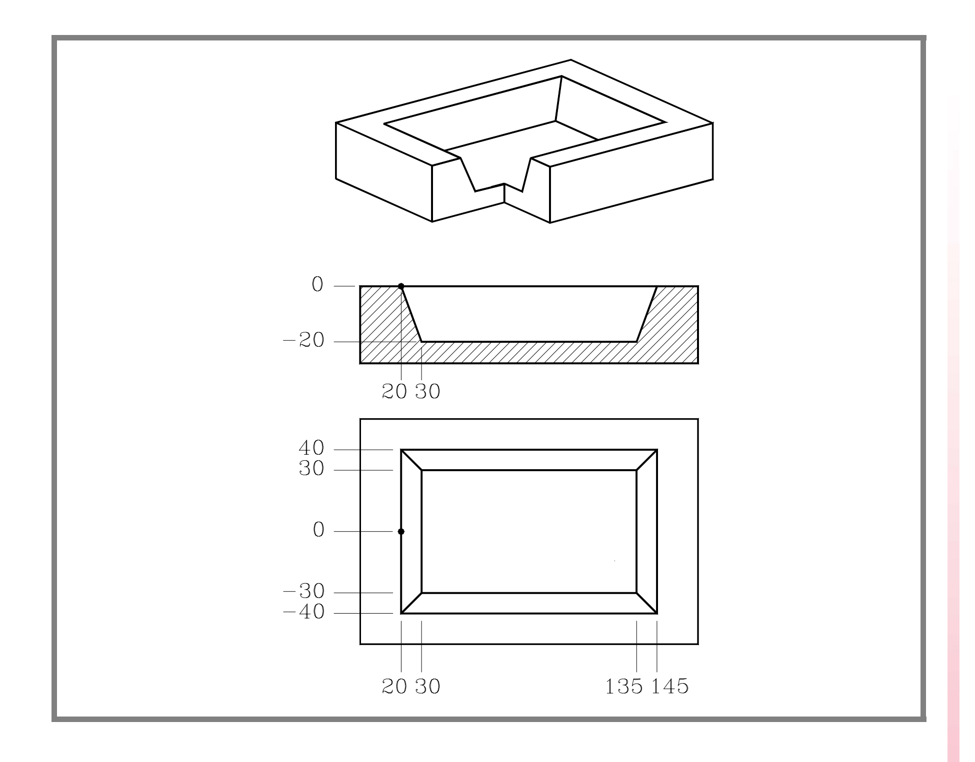
10. 精加工侧壁面。

以 "ε" 确定的铣削宽度，图标确定的加工方向执行。

快速退回 (G0) 到安全平面 (Zs)。



12.14.3 如何定义 3D 轮廓的实例



Pocket P.3D	FAGOR-A	

Profile P.XY	FAGOR 110	[调用]

设置 :

横坐标轴 : X	纵坐标轴 : Y
自动缩放 : Yes	确认

轮廓 (外轮廓):

起点	X 20	Y 0	确认
直线	X 20	Y -40	确认
直线	X 145	Y -40	确认
直线	X 145	Y 40	确认
直线	X 20	Y 40	确认
直线	X 20	Y 0	确认

结束 :

保存轮廓



Profile P.Z1	FAGOR 211	调用

设置 :

横坐标轴 : X	纵坐标轴 : Z
自动缩放 : Yes	确认

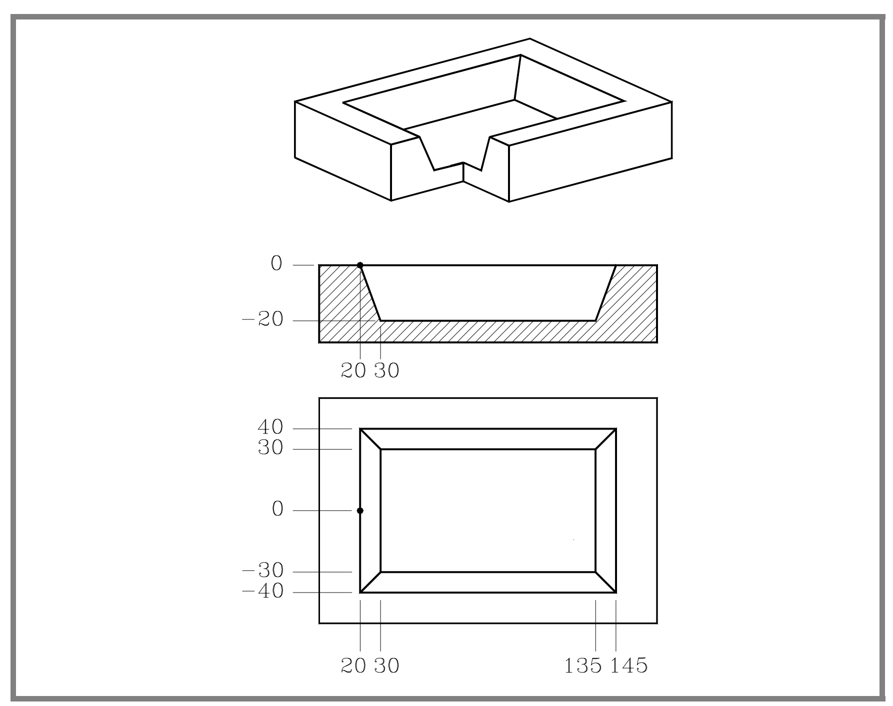
轮廓 (深度轮廓):

起点	X 20	Z0	确认
直线	X 30	Z -20	确认

结束 :

保存轮廓





Pocket P.3D	FAGOR-B	
Profile P.XY	FAGOR 120	[调用]

设置 :

横坐标轴 : X	纵坐标轴 : Y
自动缩放 : Yes	确认

轮廓 (外轮廓):

起点	X 20	Y 0	确认
直线	X 20	Y -40	确认
直线	X 145	Y -40	确认
直线	X 145	Y 40	确认
直线	X 20	Y 40	确认
直线	X 20	Y 0	确认

新轮廓 (岛):

圆	X 62.5	Y0	Xc 82.5	Yc 0	确认

结束 :

保存轮廓



Profile P.Z1	FAGOR 221	[调用]

设置 :

横坐标轴 : X	纵坐标轴 : Z
自动缩放 : Yes	确认

轮廓 (外深度轮廓):

起点	X 20	Z 0	确认
直线	X 30	Z -20	确认

结束 :

保存轮廓

Profile P.Z2	FAGOR 222	调用

设置 :

横坐标轴 : X	纵坐标轴 : Z
Autozoom: Yes	确认

轮廓 (岛深度轮廓):

起点	X 62.5	Z -20	确认
直线	X 77.5	Z 0	确认

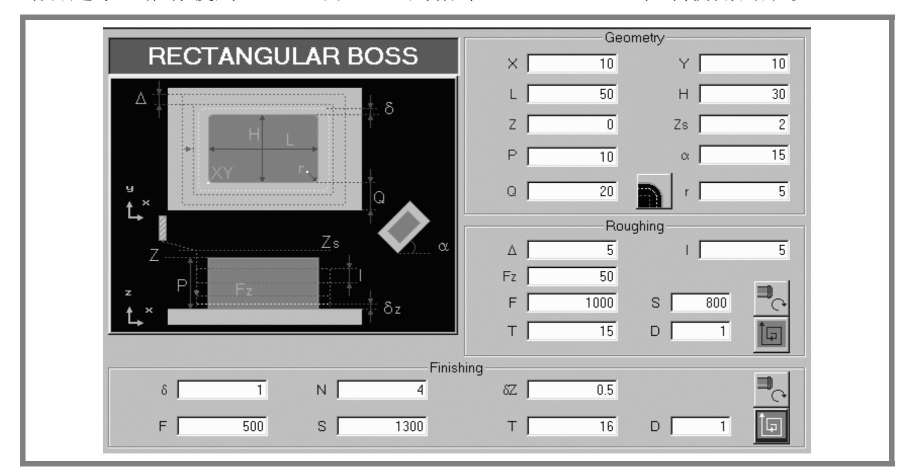
结束 :

保存轮廓

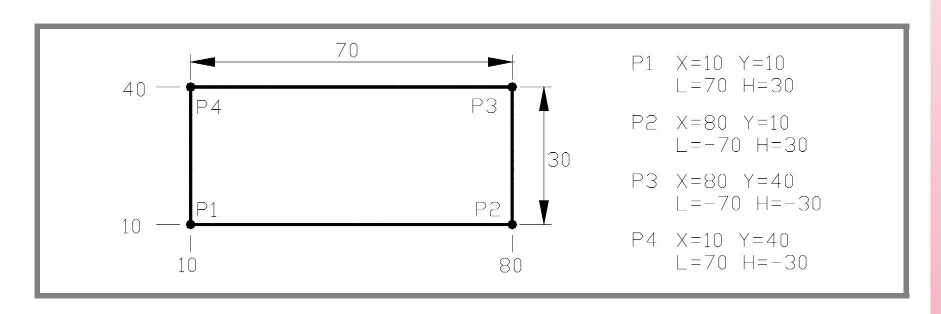


12.15 矩形凸台加工

用户预先用指令 #HSC, G5, G50 o 或 G7 设置，循环的粗加工和精加工程序段可以结合起来。推荐使用 #HSC 或 G5 ，用指令 #ROUNDPAR 控制拐角的形状。



几何参数 :



X, Y 凸台角。

L, H 凸台尺寸。

符号表示相对于 XY 点的方向。

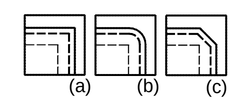
Z 工件表面坐标。

Zs 安全平面坐标。

P 总加工深度。

το 凸台与横坐标所成的角度，单位为度。旋转在 X,Y 点定义的角执行。

Q 要去除的加工总余量。



拐角类型 (图标)。

方角使用图标 (a)。

圆角使用图标 (b)。

倒角使用图标 (c)。

r 圆弧半径或倒角尺寸。

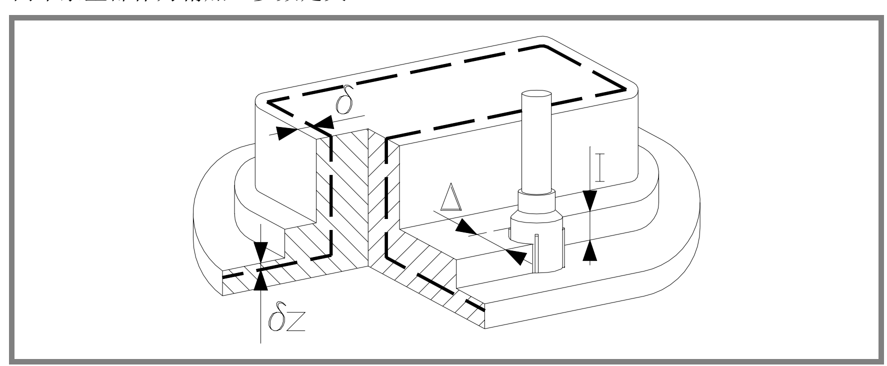
粗加工参数 :

粗加工操作加工凸台后留下如下精加工余量 :

δ 凸台侧壁面精加工余量。

δz 凸台底面精加工余量。

两个余量都作为精加工参数定义。



粗加工定义参数如下：

Δ 最大铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。

如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

I 每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号(I-),型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制单步进给量小于刀具表中所赋的切削长度值。

Fz 进刀进给率。

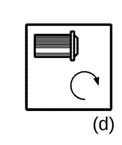
F 表面铣削进给率。

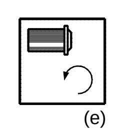
S 主轴速度。

T 粗加工刀具。

如果编写 T=0, 没有粗加工。

D 刀具偏置。

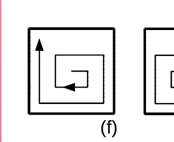

(d)


(e)

主轴旋转方向 (图标)。

顺时针方向使用图标 (d)。

逆时针方向使用图标 (e)。


(f) (g)

加工方向 (图标).

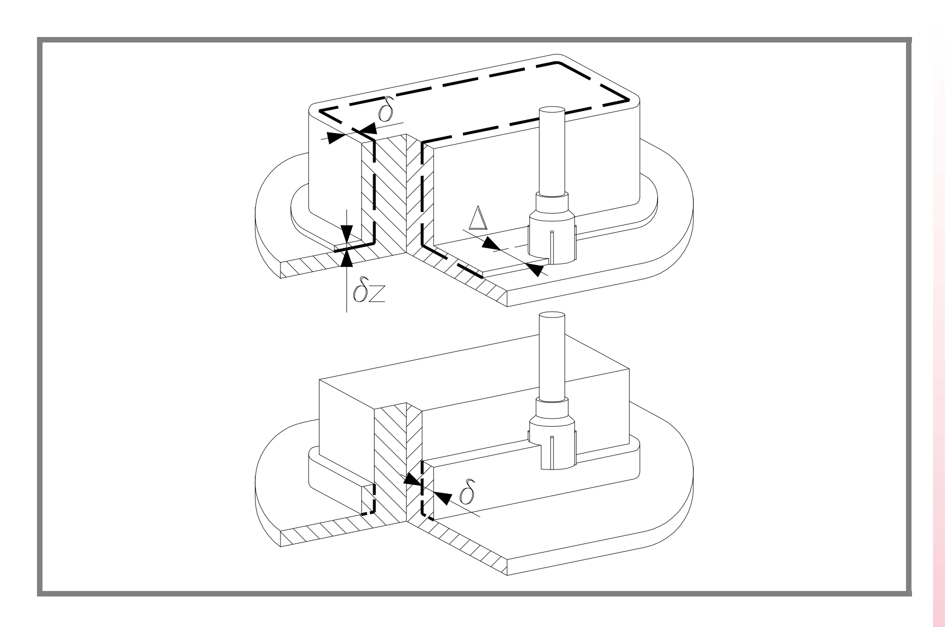
顺时针方向使用图标 (f)。

逆时针方向使用图标 (g)。

精加工参数：

精加工操作分两个阶段执行。

首先，加工凸台的底面，然后切向进入和退出加工侧壁面。



精加工定义参数如下 :

δ 凸台侧壁面精加工余量。

δz 凸台底面精加工余量。

N 侧面精加工的进刀的数量（步数）。如果作为结果的一步大于刀具表中赋予刀具的切削长度， 步长受表中值的限制。

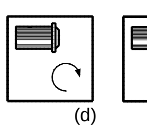
F 表面和侧壁面铣削进给率。

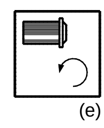
S 主轴速度。

T 精加工刀具。

如果编写 T=0, 没有精加工。

D 刀具偏置。


(d)

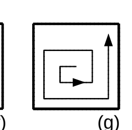

(e)

主轴旋转方向 (图标)。

顺时针方向使用图标 (d)。

逆时针方向使用图标 (e)。

(f)


(g)

加工方向 (图标)。

顺时针方向使用图标 (f)。

逆时针方向使用图标 (g)。



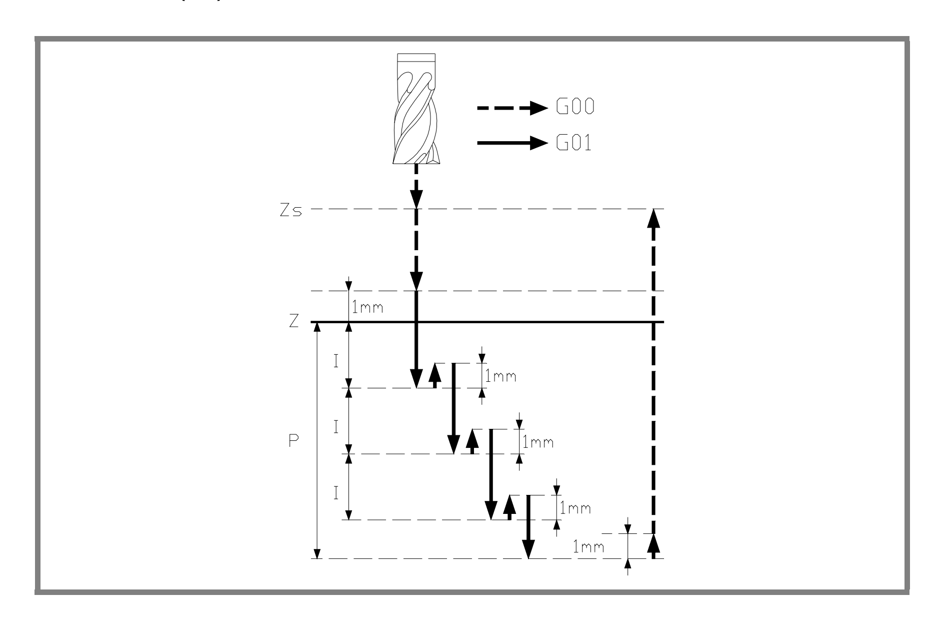
12.15.1 基本操作 :

1. 选择粗加工刀具，启动主轴以需要的方向旋转。

2. 快速运动 (G0) 到粗加工起点和安全平面 (Zs)。

依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。

3. 快速运动 (G0) 到接近平面。



4. 粗加工操作。

以层加工，直到到达总加工深度减去底面精加工余量 "δz"。

·1· 以进给率 "Fz" 进刀 "I" 。

·2· 铣削凸台表面直到距型腔壁"δ"的距离。以进给率"F"执行，如果需要，重新计算最大切削宽度 (Δ)，这样所有的切削宽度一致。

·3· 快速退回 (G0) 到起点。

5. 快速退回 (G0) 到安全平面 (Zs)。

6. 选择精加工刀具，快速接近距最后一次粗加工操作 1 mm 处 。

7. 精加工凸台底面。

·1· 以进给率 "Fz" 进刀。

·2· 铣削凸台底面直到距侧壁面 "δ" 的距离。以精加工进给率 "F" 和粗加工铣削宽度执行 It is carried out at the finishing feedrate "F" and with the roughing pass.

8. 快速退回 (G0) 到在接近平面的起点 。

9. 精加工侧壁面。

以进给率 "F" ， 切入和退出，执行 "N" 次进刀。

10.快速退回 (G0) 到安全平面 (Zs)。

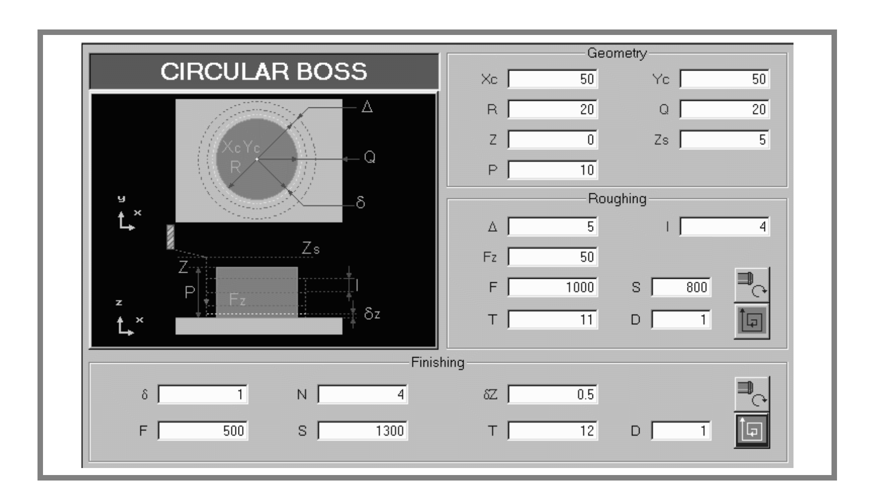
如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

11.快速运动 (G0) 到下一点 。

12.重复步骤 3, 4, 5, 6, 7,8 , 9, 10。



12.16 圆形凸台加工



几何参数 :

Xc, Yc　凸台中心。

R　　凸台半径。

Z　　工件表面坐标。

Zs　　安全平面坐标。

P　　总加工深度。

Q　　要去除的加工总余量。

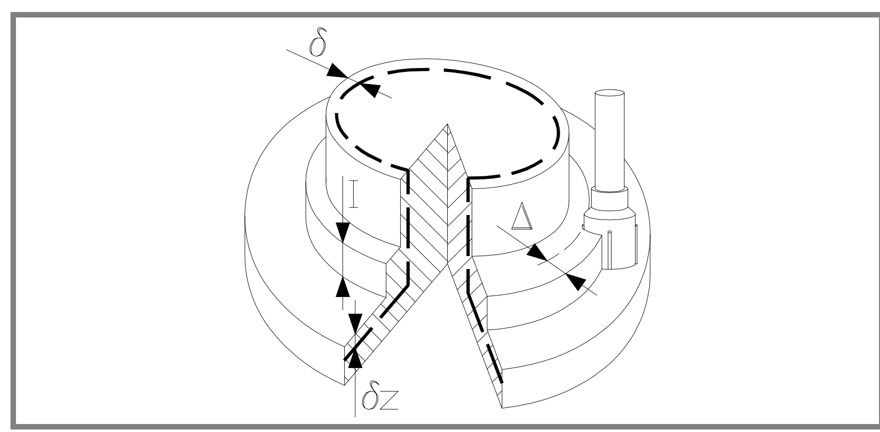
粗加工参数 :

粗加工操作加工凸台留下如下精加工余量 :

δ　　凸台侧壁面精加工余量。

δz　　凸台底面精加工余量。

两个余量都作为精加工参数定义。



粗加工操作定义参数如下 :

Δ　　最大铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。



I　　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号(I-),型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制每步进给量小于刀具表中所赋的切削长度值。

Fz　　进刀进给率。

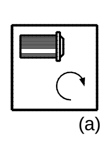
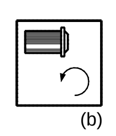
F　　表面铣削进给率。

S　　主轴速度。

T　　粗加工刀具。

如果编写 T=0, 没有粗加工。

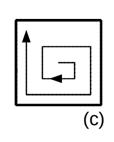
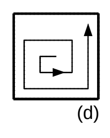
D　　刀具偏置。


(a)

(b)

主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。


(c)

(d)

加工方向 (图标)。

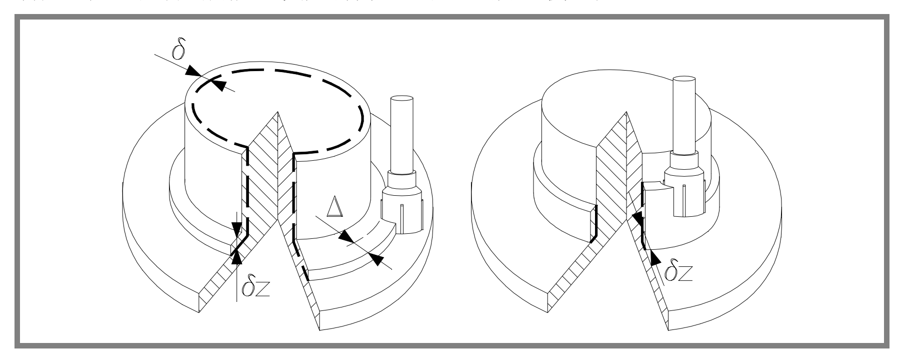
顺时针方向使用图标 (c)。

逆时针方向使用图标 (d)。

精加工参数 :

精加工操作分两个阶段执行。

首先，加工凸台的底面，然后切向进入和退出加工侧壁面。



精加工操作定义参数如下 :

δ 凸台侧壁面精加工余量。

δz 凸台底面精加工余量。

N 侧面精加工的进刀的数量（步数）。如果作为结果的一步大于刀具表中赋予刀具的切削长度， 步长受表中值的限制。

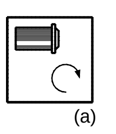
F 表面和侧壁面铣削进给率。

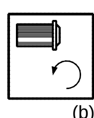
S 主轴速度。

T 精加工刀具。

如果编写 T=0, 没有精加工 。

D 刀具偏置。

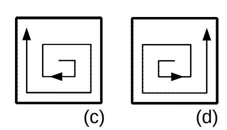




主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。



加工方向 (图标).

顺时针方向使用图标 (c)。

逆时针方向使用图标 (d)。



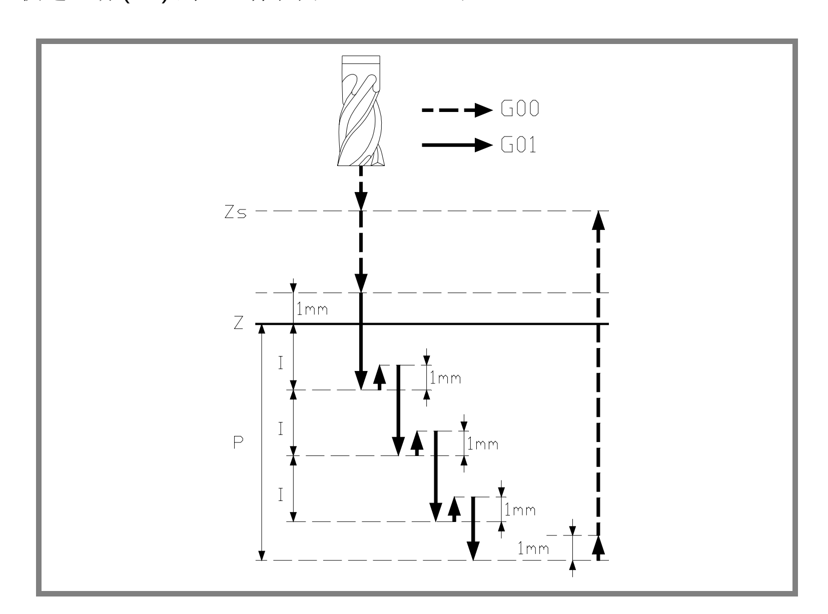
12.16.1 基本操作

1. 选择粗加工刀具，启动主轴以需要的方向旋转。

2. 快速运动 (G0) 到粗加工起点和安全平面 (Zs)。

依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。

3. 快速运动 (G0) 到距工件表面 "Z" 上 1 mm 处。



4. 粗加工操作。

以层加工，直到到达总加工深度减去底面精加工余量 "δz"。

·1· 以进给率 "Fz" 进刀 "I" 。

·2· 铣削凸台表面直到距型腔壁"δ"的距离。以进给率"F"执行，如果需要，重新计算最大切削宽度 (Δ) ，这样所有的切削宽度一致。

·3· 快速退回 (G0) 到起点。

5. 快速退回 (G0) 到安全平面 (Zs)。

6. 选择精加工刀具，快速接近距最后一次粗加工操作 1 mm 处 。

7. 精加工凸台底面。

·1· 以进给率 "Fz" 进刀。

·2· 铣削凸台底面直到距侧壁面 "δ" 的距离。以精加工进给率 "F" 和粗加工铣削宽度执行。

8. 快速退回 (G0) 到接近平面上的起点。

9. 精加工侧壁面。

以进给率 "F" ， 切入和退出，执行 "N" 次进刀。

10.快速退回 (G0) 到安全平面 (Zs)。

如果和多重加工操作相关联，将尽可能频繁的执行下列步骤：

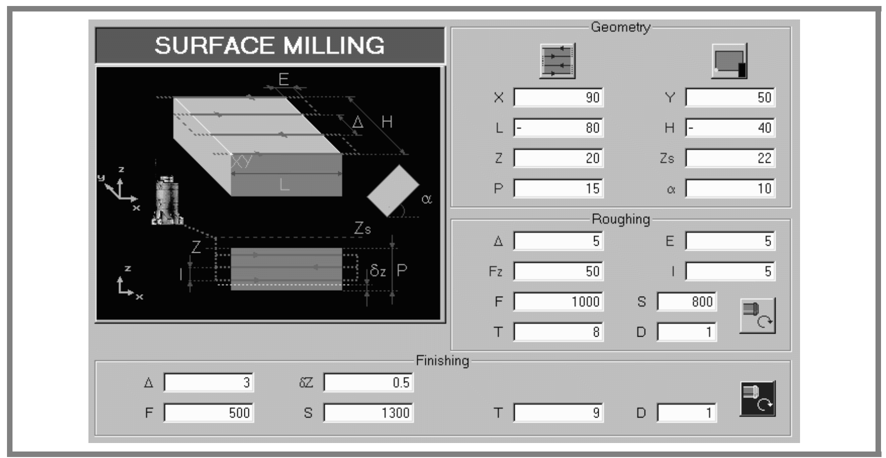
11.快速运动 (G0) 到下一点 。

12.重复步骤 3, 4, 5, 6, 7,8 , 9, 10。



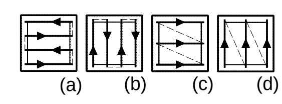
12.17 表面铣削

用户预先用指令 #HSC, G5, G50 o 或 G7 设置，循环的粗加工和精加工程序段可以被结合起来。推荐使用 #HSC 或 G5 ，用指令 #ROUNDPAR 控制拐角的形状。



几何参数 :

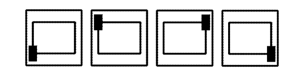
加工方向 (图标).



X 轴双向 (a), Y 轴双向 (b)。

X 轴单向 (c), Y 轴单向 (d)。

表面铣削起始角 (图标).



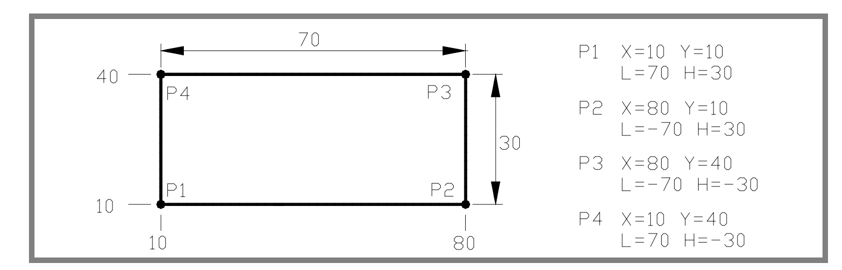
个可选角中任意一种。

X, Y, L, H

被铣削表面。

定义要铣削的表面的一个角 (X, Y), 长度 (L) 和宽度 (H) 。

点 (X, Y) 不必和选择的加工起点的角一致。L 和 H 的符号表示其相对 XY 点的方向。



Z 工件表面坐标。

Zs 安全平面坐标。

P 总加工深度。

το 表面和横坐标轴之间的角，单位度。旋转在 X,Y 点定义的角执行。



粗加工参数 :

粗加工操作留下精加工余量 δz，其作为精加工参数被定义。

粗加工操作定义参数如下 :

Δ　最大铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

E　刀具运动超出铣削表面的距离。

Fz　进刀进给率。

I　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号 (I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制每步进给量小于刀具表中所赋的切削长度值。

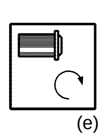
F　表面铣削进给率。

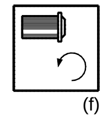
S　主轴速度。

T　粗加工刀具。

如果编写 T=0, 没有粗加工。

D　刀具偏置。


(e)


(f)

主轴旋转方向 (图标)。

顺时针方向使用图标 (e)。

逆时针方向使用图标 (f)。

精加工参数 :

δz　精加工余量。

Δ　最大铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

F　表面铣削进给率。

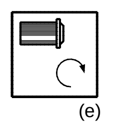
S　主轴速度。

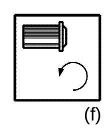


T 粗加工刀具。

如果编写 T=0, 没有粗加工。

D 刀具偏置。





主轴旋转方向 (图标)。

顺时针方向使用图标 (e)。

逆时针方向使用图标 (f)。

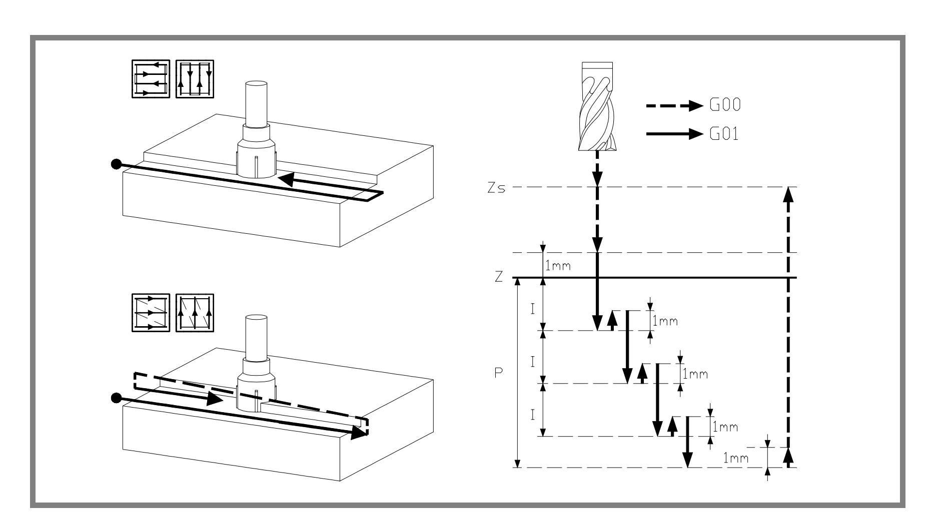


12.17.1 基本操作

1. 选择粗加工刀具，启动主轴以需要的方向旋转。

2. 快速运动 (G0) 到粗加工起点和安全平面 (Zs)。

依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。



3. 快速运动 (G0) 到接近平面。

4. 粗加工操作。

以层加工，直到到达总加工深度减去底面精加工余量 "δz"。

·1· 以进给率 "Fz" 进刀 "I" 。

·2· 以进给率 "F" 铣削，如果需要，重新计算最大切削宽度 (Δ)，这样所有的切削宽度一致。

在双向铣削方式，所有运动以进给率 "F"。

在单向铣削方式，在两个连续铣削路径中间的运动以快速方式并在工件表面上方 1 mm 处 执行。

·3· 快速退回 (G0) 到工件上方 1 mm。

·4· 快速运动 (G0) 到起点。

5. 快速退回 (G0) 到安全平面 (Zs)。

6. 精加工操作 .

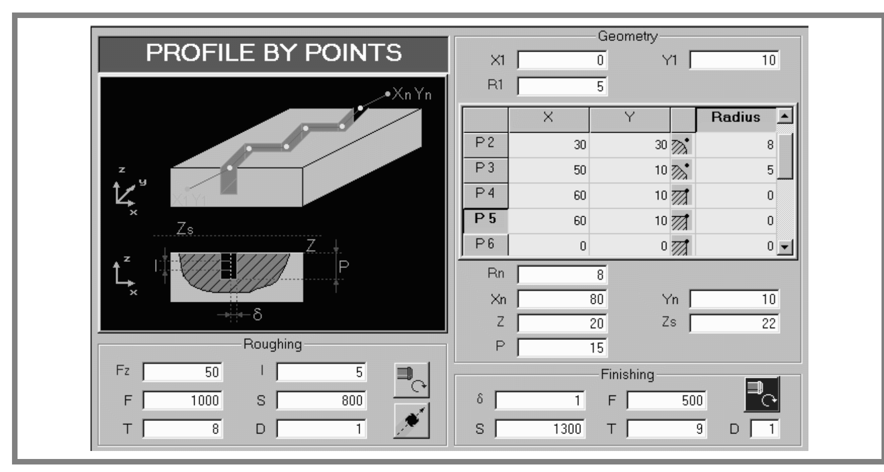
·1· 以进给率 "Fz" 进刀。

·2· 以进给率 "F" 铣削，如果需要，重新计算最大切削宽度 (Δ)，这样所有的切削宽度一致。

7. 快速退回 (G0) 到安全平面 (Zs)。

12.18 点到点 轮廓

用户预先用指令 #HSC, G5, G50 o 或 G7 设置，循环的粗加工和精加工程序段可以结合起来。推荐使用 #HSC 或 G5 ，用指令 #ROUNDPAR 控制拐角的形状。

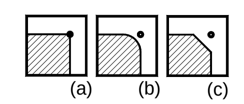


几何参数 :

X1, Y1　　轮廓进入点。

R1　　　切入轮廓的半径。

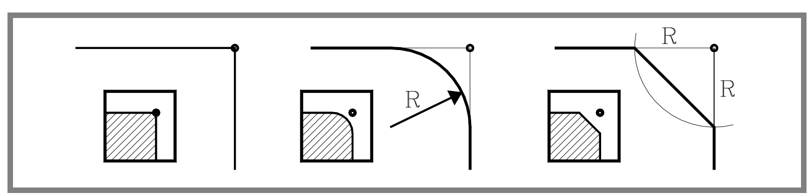
P1..P12　轮廓的点。



所有中间点 P2 到 P11 有一个说明拐角类型的图标 方形(a), 圆弧(b) 或斜边(c)。

对于圆弧或斜边拐角，指出圆弧半径或斜边尺寸。

当没有使用全部的 12 个点时，用和轮廓最后一个点相同的坐标定义第一个不使用的点 。



Rn　　切出轮廓的半径。

Xn, Yn　轮廓出口点。

Z　　　工件表面坐标。

Zs　　　安全平面坐标。

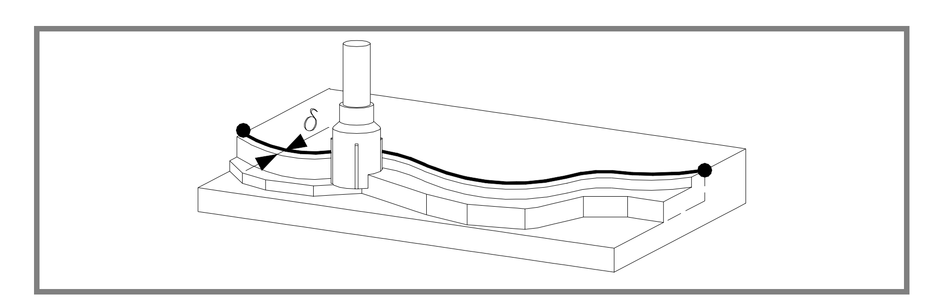
P　　　总加工深度。



粗加工参数 :

粗加工操作铣削轮廓，留下精加工余量 δ。

该余量作为精加工参数定义。



粗加工操作定义参数如下 :

Fz　　进刀进给率。

I　　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号 (I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制每步进给量小于刀具表中所赋的切削长度值。

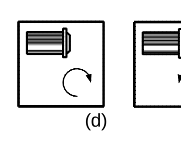
F　　表面铣削进给率。

S　　主轴速度。

T　　粗加工刀具。

如果编写 T=0, 没有粗加工。

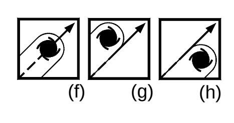
D　　刀具偏置。



主轴旋转方向 (图标)。

顺时针方向使用图标 (d)。

逆时针方向使用图标 (e)。



刀具半径补偿 (图标)。

无刀具半径补偿 (f)。

左刀具半径补偿 (g)。

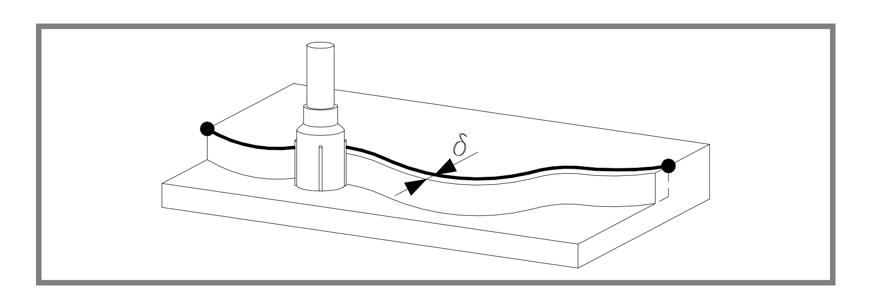
右刀具半径补偿 (h)。



精加工参数 :

为了执行精加工操作，粗加工必须定义刀具半径补偿。

该操作去除精加工余量 (δ)。



粗加工操作定义参数如下 :

δ 侧壁面精加工余量。

当在无刀具半径补偿方式下加工时，没有精加工，精加工余量 (δ) 被忽略 。

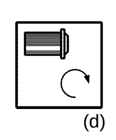
F 铣削进给率。

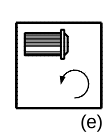
S 主轴速度。

T 精加工刀具。

如果编写 T=0, 没有精加工。

D 刀具偏置。


(d)


(e)

主轴旋转方向 (图标)。

顺时针方向使用图标 (d)。

逆时针方向使用图标 (e)。



12.18.1 基本操作

1. 选择粗加工刀具，启动主轴以需要的方向旋转。

2. 快速运动 (G0) 到粗加工起点和安全平面 (Zs)。依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。

3. 快速运动 (G0) 到接近平面。

4. 粗加工操作。

以层加工，直到到达总加工深度。

·1· 以进给率 "Fz" 进刀 "I" 。

·2· 以进给率 "F" 铣削轮廓，切入，如果编写了。

如果定义了刀具半径补偿，距壁面 "δ" 的距离执行铣削。

·3· 切出点 XnYn，如果编写了 。

·4· 快速退回 (G0) 到安全平面 (Zs)。

·5· 快速运动到起点 X1Y1。

5. 选择精加工刀具，启动主轴以需要的方向旋转。

6. 精加工操作。

7. 以进给率 "Fz" 进刀底面。

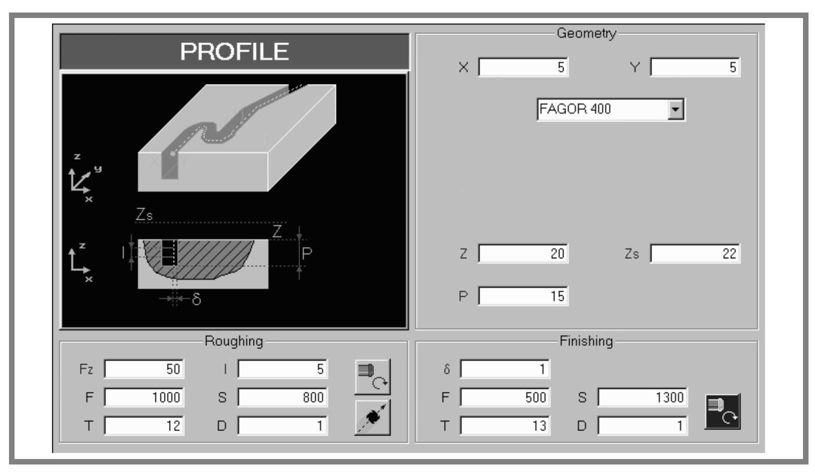
·1· 以进给率 "F" 铣削轮廓，如果编写了则以切线方式进入。

·2· 如果编写了则以切线方式退出点 XnYn 。

8. 快速退回 (G0) 到安全平面 (Zs)。

12.19 轮廓

用户预先用指令 #HSC, G5, G50 o 或 G7 设置，循环的粗加工和精加工程序段可以结合起来。推荐使用 #HSC 或 G5 ，用指令 #ROUNDPAR 控制拐角的形状。



几何参数 :

X, Y　　轮廓起点。

轮廓名字。

为以切入和退出方式加工，定义轮廓内的这些值。

Z　　工件表面坐标。

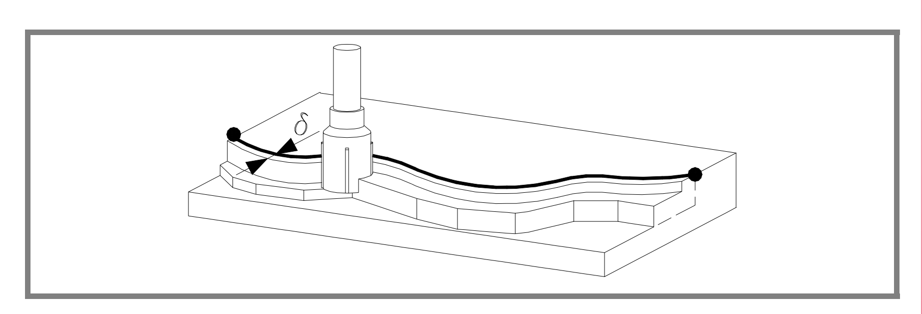
Zs　　安全平面坐标。

P　　总加工深度。

粗加工参数 :

粗加工操作铣削留下精加工余量 δ。

该余量作为精加工参数定义。





粗加工操作定义参数如下 :

Fz　进刀进给率。

I　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号(I-),型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制每步进给量小于刀具表中所赋的切削长度值。

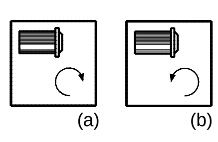
F　表面铣削进给率。

S　主轴速度。

T　粗加工刀具。

如果编写 T=0, 没有粗加工。

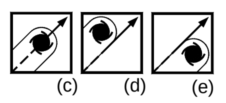
D　刀具偏置。



主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。



刀具半径补偿 (图标)。

无刀具半径补偿 (c)。

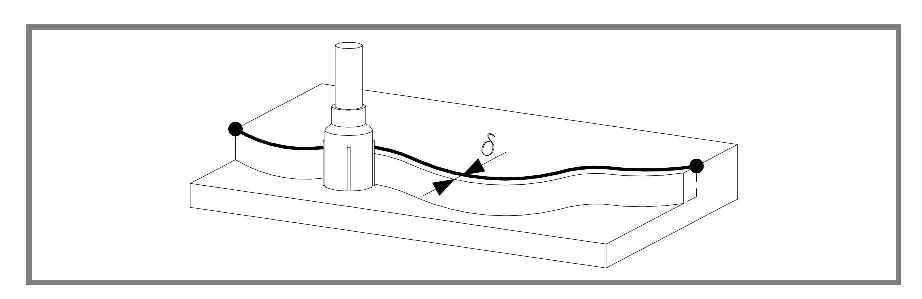
左刀具半径补偿 (d)。

右刀具半径补偿 (e)。

精加工参数 :

为了执行精加工操作，粗加工必须定义刀具半径补偿。

该操作去除精加工余量 (δ)。



粗加工定义参数如下 :

δ　侧壁面精加工余量。

当在无刀具半径补偿方式下加工时，没有精加工，精加工余量 (δ) 被忽略。这种情况时，精加工同粗加工时的刀具中心运动。

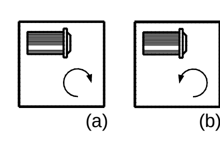
F　铣削进给率。

S　主轴速度。

T 精加工刀具。

如果编写 T=0, 没有精加工。

D 刀具偏置。



主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。



12.19.1 基本操作

1. 选择粗加工刀具，启动主轴以需要的方向旋转。
2. 快速运动 (G0) 到粗加工起点和安全平面 (Zs)。依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。
3. 快速运动 (G0) 到接近平面。
4. 粗加工操作。

   以层加工，直到到达总加工深度。

   ·1· 以进给率 "Fz" 进刀 "I" 。

   ·2· 以进给率 "F" 铣削轮廓。

   如果定义了刀具半径补偿，距壁面 "δ" 的距离执行铣削。

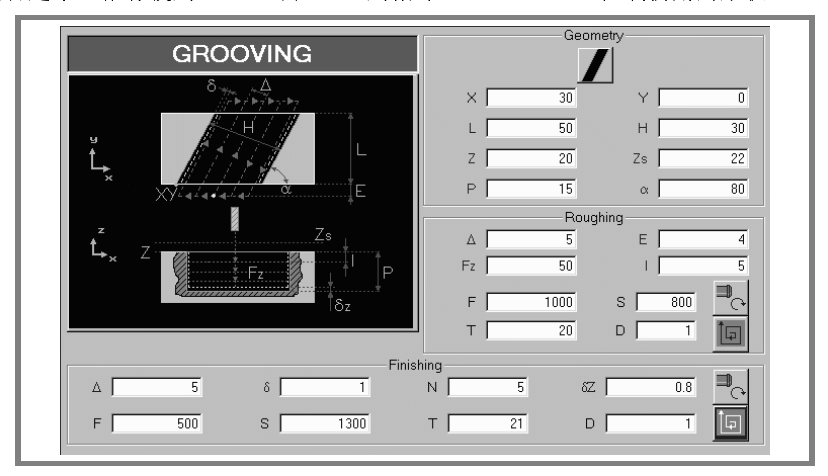
   ·3· 快速退回 (G0) 到安全平面 (Zs)。

   ·4· 快速运动到起点 X1Y1。
5. 选择精加工刀具，启动主轴以需要的方向旋转。
6. 精加工操作。
7. 以进给率 "Fz" 进刀底面。
   - 以进给率 "F" 铣削轮廓。
8. 快速退回 (G0) 到安全平面 (Zs)。



12.20 狭槽铣削

用户预先用指令 #HSC, G5, G50 o 或 G7 设置，循环的粗加工和精加工程序段可以结合起来。推荐使用 #HSC 或 G5 ，用指令 #ROUNDPAR 控制拐角的形状。



几何参数 :

狭槽铣削类型 (图标).

有六种可能的类型。

4 种是铣削工件每个拐角旁的狭槽。

2 种是铣削穿过工件的狭槽。

X, Y　铣削狭槽的角。

L, H　狭槽尺寸。

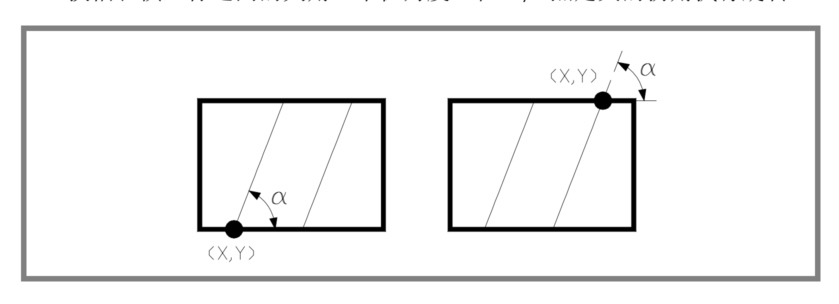
符号表示相对于 XY 点的方向。

Z　工件表面坐标。

Zs　安全平面坐标。

P　总加工深度。

τo　狭槽和横坐标之间的夹角，单位为度。在 X,Y 点定义的拐角执行旋转。





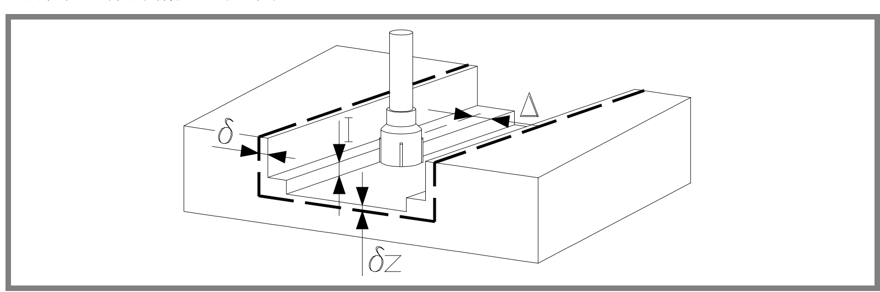
粗加工参数 :

粗加工操作留下如下精加工余量 :

δ　　侧壁面精加工余量。

δz　　狭槽底面精加工余量。

两种余量作为精加工参数定义。



粗加工操作定义参数如下 :

Δ　　最大铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

E　　刀具运动超出铣削表面的距离。

Fz　　进刀进给率。

I　　每步进刀量。

- 如果编写正号 (I+), 循环重新计算，所有每步进刀量一致，用相同的值或者小于编程值。
- 如果编写负号(I-), 型腔加工以给定的每步进刀量加工，除了最后一步进刀量是加工余量 。

在任何一种情况，循环限制每步进给量小于刀具表中所赋的切削长度值。

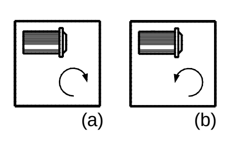
F　　表面铣削进给率。

S　　主轴速度。

T　　粗加工刀具。

如果编写 T=0, 没有粗加工。

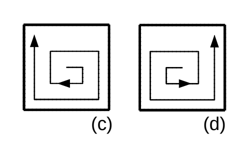
D　　刀具偏置。



主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。



加工方向 (图标)。

顺时针方向使用图标 (c)。

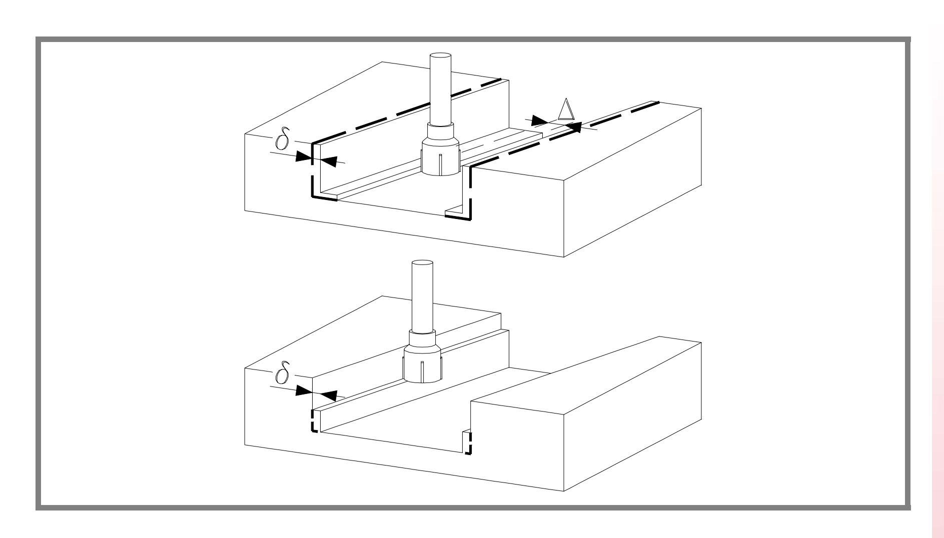
逆时针方向使用图标 (d)。



精加工参数 :

精加工操作分两个阶段执行。

首先，加工狭槽的底面，然后切向进入和退出加工侧壁面。



精加工操作定义参数如下 :

δ 侧壁面精加工宽度。

δz 底面精加工宽度。

Δ 狭槽底面铣削宽度。

循环重新计算铣削宽度，这样所有加工宽度一致，用相同的值或是小于编程的值。
如果编程值为 0 值 , 采用所选刀具直径值的 3/4 。

N 侧面精加工的进刀的数量（步数）。如果作为结果的一步大于刀具表中赋予刀具的切削长度 ， 步长受表中值的限制。

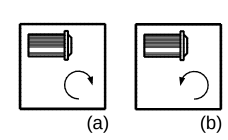
F 表面和侧壁面铣削进给率。

S 主轴速度。

T 精加工刀具。

如果编写 T=0, 没有精加工。

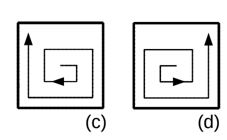
D 刀具偏置。



主轴旋转方向 (图标)。

顺时针方向使用图标 (a)。

逆时针方向使用图标 (b)。



加工方向 (图标).

顺时针方向使用图标 (c)。

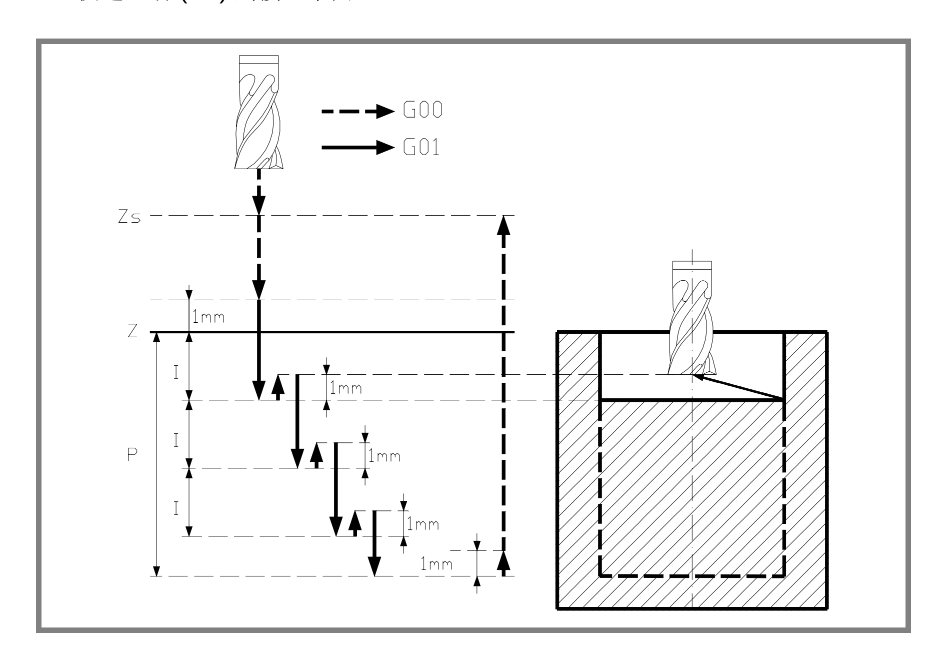
逆时针方向使用图标 (d)。



12.20.1 基本操作

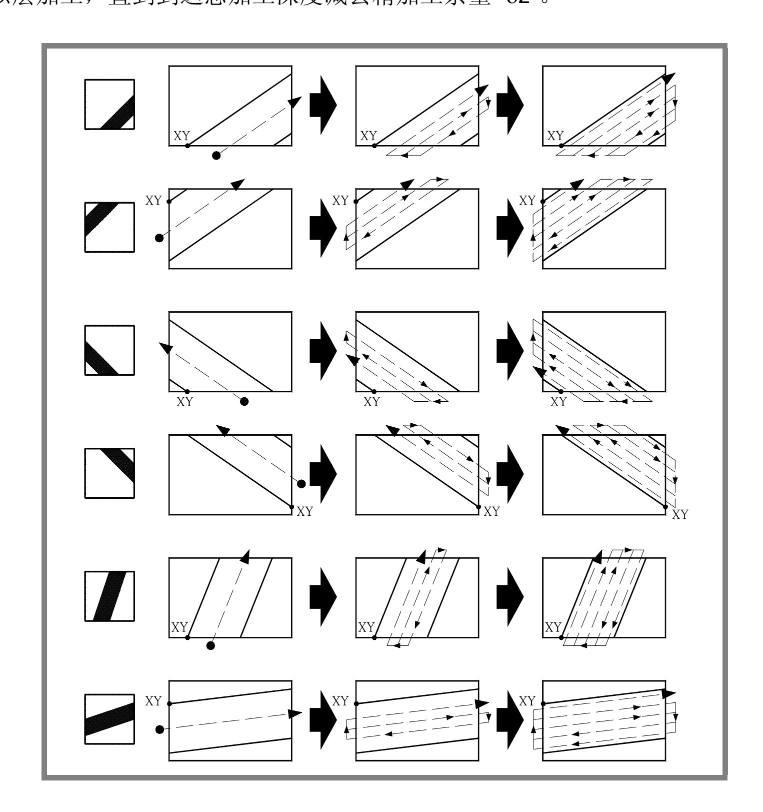
1. 选择粗加工刀具，启动主轴以需要的方向旋转。
2. 快速运动 (G0) 到粗加工起点和安全平面 (Zs)。

   依据初始平面的位置，先在 XY 平面运动，然后沿 Z 轴运动。反之亦然 。
3. 快速运动 (G0) 到接近平面。



4. 粗加工操作。

   以层加工，直到到达总加工深度减去精加工余量 "δz"。



·1· 以进给率 "Fz" 进刀 "I" 。

·2· 铣削狭槽直到距侧壁面"δ"。以进给率"F"执行，如果需要，可重新计算铣削宽度 (Δ) 使所有铣削宽度一致。

·3· 快速退回 (G0) 到安全平面 (Zs)。

·4· 快速运动 (G0) 到起点。

·5· 快速运动 (G0) 到距已加工表面上方 1 mm 处。



5. 快速退回 (G0) 到安全平面 (Zs)。

6. 选择精加工刀具，快速 (G0) 接近到距粗加工完成底面 1 mm 处。

7. 精加工狭槽底面

·1· 以 "Fz" 进给率进刀。

·2· 铣削狭槽底面直到距狭槽侧壁面 "δ" 。以精加工进给率 "F" 执行，如果需要，可重新计算铣削宽度 (Δ) 使所有铣削宽度一致。

8. 快速退回 (G0) 到安全平面 (Zs)。

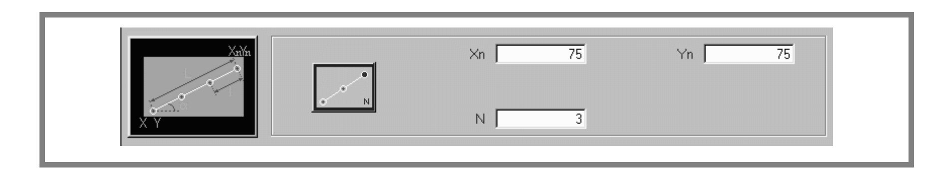
9. 精加工侧壁面。

以精加工进给率 "F" 执行 "N" 次进刀加工。

10. 快速退回 (G0) 到安全平面 (Zs)。



12.21 在直线模式上的多重加工





定义格式 (图标)。

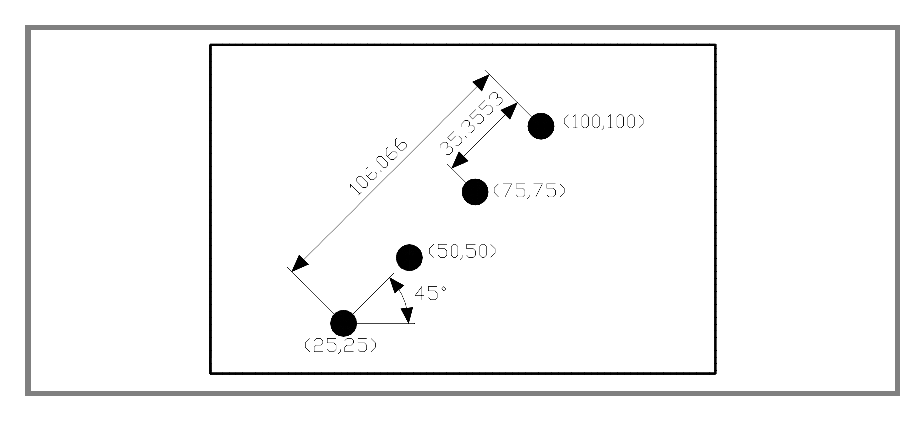
有 5 种不同的方法定义这种加工操作。

要选择想要的一种，将光标移到图标上然后按下空格键。

加工操作数目 "N" 也必须包括循环定义点的操作。

编程实例 :

固定循环定义在点 X25, Y25，在其余的点重复。



现在说明 5 种定义它的可能的方法。

1)	终点坐标	Xn 100, Yn 100
	加工操作的总数目	N 4
2)	路径的转角	α 45
	路径长度	L 106.066
	加工操作的总数目	N 4
3)	路径的转角	α 45
	加工操作的总数目	N 4
	两个加工操作之间的距离	I 35.3553
4)	终点坐标	Xn 100, Yn 100
	两个加工操作之间的距离	I 35.3553
5)	路径的转角	α 45
	路径长度	L 106.066
	两个加工操作之间的距离	I 35.3553



12.22 在圆弧模式上的多重加工

定义格式 (图标)。

有 9 种不同的方法定义这种加工操作。

要选择想要的一种，将光标移到图标上然后按下空格键。

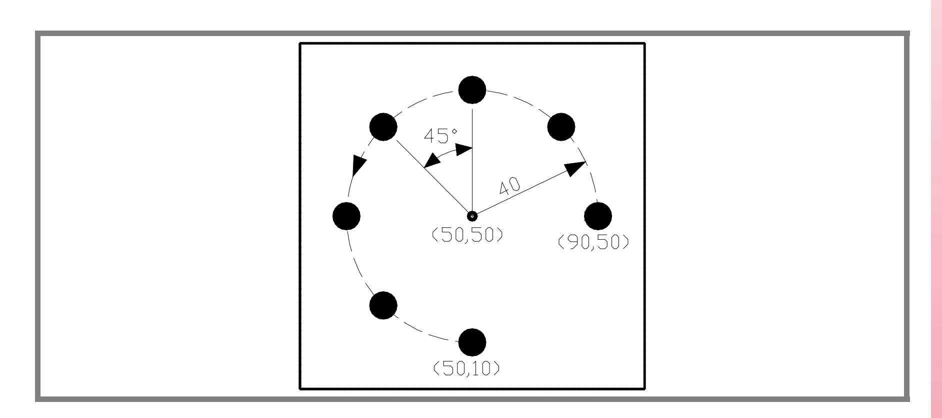
加工操作数目 "N" 也必须包括为定义点的循环。

圆弧上的运动逆时针方向进行。要执行顺时针方向，定义两个操作之间的夹角 β，带负号表示。

加工操作数目 "N" 必须包含循环定义点的操作。

编程实例：

固定循环定义在 X90, Y50，在其余点重复。



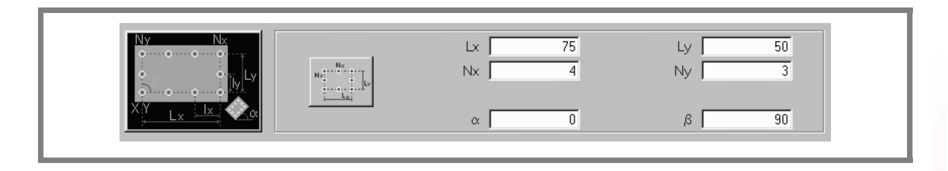
现在说明 9 种可以定义它的方法。

1)	中心坐标	Xa 50, Ya 50
	加工操作的总数目	N 7
	终点的角度	τ 270
2)	中心坐标	Xa 50, Ya 50
	加工操作的总数目	N 7
	两加工操作之间的角距离	β 45
3)	半径	R 40
	加工操作的总数目	N 7
	起点的角度	α 0
	终点的角度	τ 270
4)	半径	R 40
	加工操作的总数目	N 7
	起点的角度	α 0
	两加工操作之间的角距离	β 45

5)	中心坐标	Xa 50, Ya 50
	终点的角度	τ 270
	两加工操作之间的角距离	β 45
6)	半径	R 40
	起点的角度	α 0
	终点的角度	τ 270
	两加工操作之间的角距离	β 45
7)	中心坐标	Xa 50, Ya 50
	半径	R 40
	加工操作的总数目	N 7
	起点的角度	α 0
	两加工操作之间的角距离	β 45
8)	中心坐标	Xa 50, Ya 50
	半径	R 40
	加工操作的总数目	N 7
	起点的角度	α 0
	终点的角度	τ 270
9)	中心坐标	Xa 50, Ya 50
	半径	R 40
	起点的角度	α 0
	终点的角度	τ 270
	两加工操作之间的角距离	β 45



12.23 在平行四边形模式上的多重加工



定义格式 (图标)。

有 3 种不同的方法定义这种加工操作。

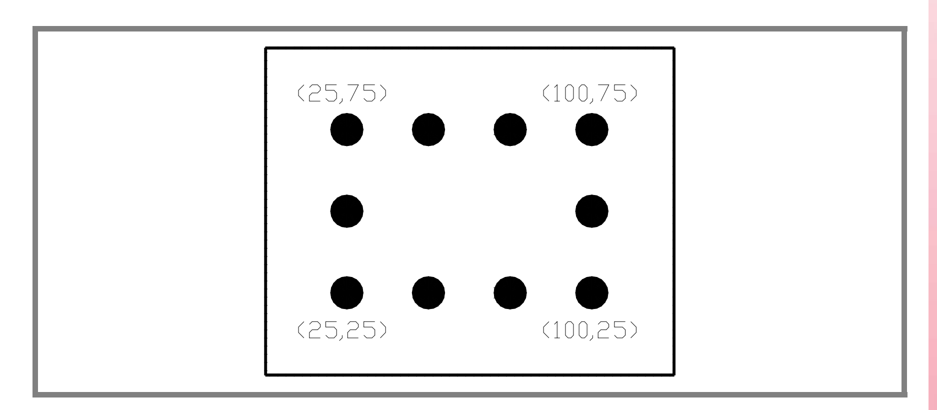
要选择想要的一种，将光标移到图标上然后按下空格键。

循环采用左下角点作为起点。如果不是，定义两孔之间的距离 Ix 和 Iy，并加上正确的符号。

加工操作数目 "N" 也必须包括为定义点的循环。

编程实例 :

固定循环定义在点 X25, Y25，在其余点重复。

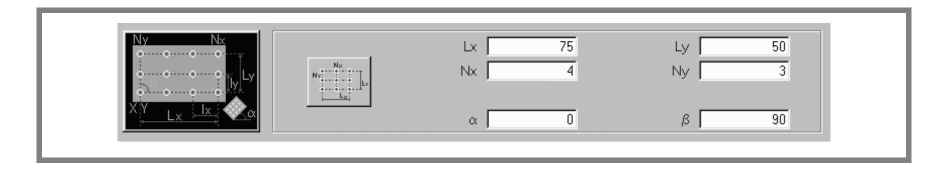


现在说明 3 种定义它的可能的方法。

1)	相对点 X,Y 的长度	Lx 75, Ly 50
	在 X 和 Y 方向的加工操作数目	Nx 4, Ny 3
	旋转角度	α 0
	两路径之间的夹角	β 90
2)	在 X 和 Y 方向的加工操作数目	Nx 4, Ny 3
	在 X 轴和 Y 轴方向的两加工操作之间的距离	Ix 25, Iy 25
	旋转角度	α 0
	两路径之间的夹角	β 90
3)	相对点 X,Y 的长度	Lx 75, Ly 50
	在 X 轴和 Y 轴方向的两加工操作之间的距离	Ix 25, Iy 25
	旋转角度	α 0
	两路径之间的夹角	β 90



12.24 在网格模式上的多重加工



定义格式 (图标)。

有 3 种不同的方法定义这种加工操作。

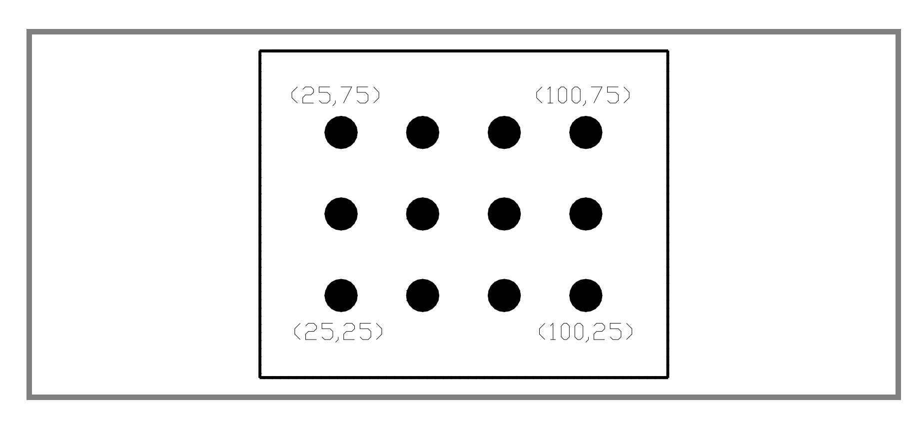
要选择想要的一种，将光标移到图标上然后按下空格键。

循环采用左下角点作为起点。如果不是，定义两孔之间的距离 Ix 和 Iy，并加上正确的符号。

加工操作数目 "N" 也必须包括为定义点的循环

编程实例 :

固定循环定义在点 X25, Y25，在其余点重复。

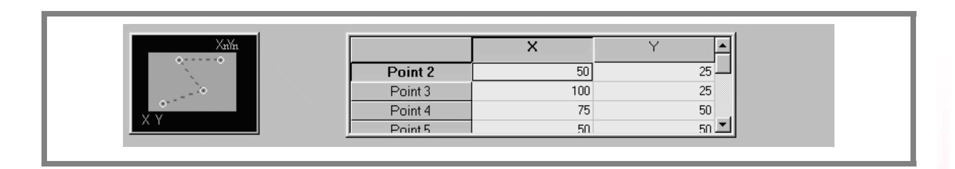


现在说明 3 种定义它的可能的方法。

1)	相对点 X,Y 的长度	Lx 75, Ly 50
	在 X 和 Y 方向的加工操作数目	Nx 4, Ny 3
	旋转角度	α 0
	两路径之间的夹角	β 90
2)	在 X 和 Y 方向的加工操作数目	Nx 4, Ny 3
	在 X 轴和 Y 轴方向的两加工操作之间的距离	Ix 25, Iy 25
	旋转角度	α 0
	两路径之间的夹角	β 90
3)	相对点 X,Y 的长度	Lx 75, Ly 50
	在 X 轴和 Y 轴方向的两加工操作之间的距离	Ix 25, Iy 25
	旋转角度	α 0
	两路径之间的夹角	β 90



12.25 任意多重加工



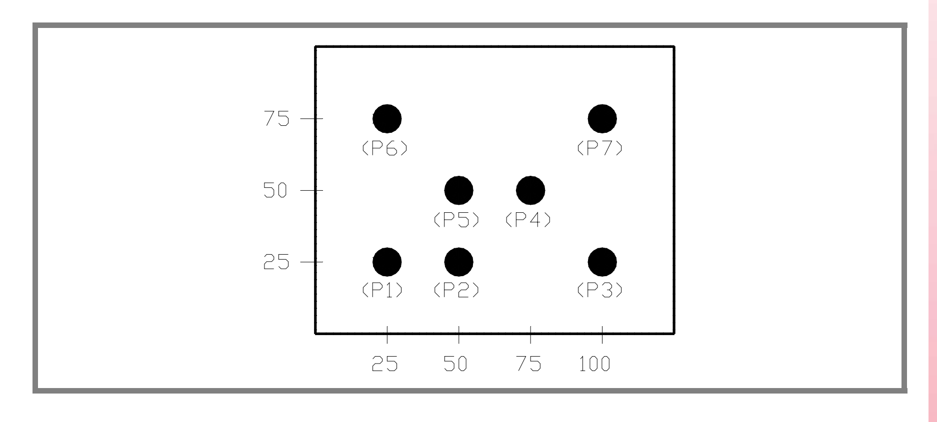
开始点是循环定义点。

其余的点 (P2) 到 (P12) 必须在多重加工区域内进行定义。

当不用所有的点时，用和轮廓最后一个点相同的坐标定义第一个不用的点。

编程实例 :

固定循环定义在点 X25, Y25，在其余点重复。



固定循环定义在点 (P1) X25, Y25。

其余的点 (P2) 到 (P7) 在多重加工区域内进行定义。

因为只有 7 个点 , 必须定义 (P8) = (P7).

(P2)	X 50	Y 25
(P3)	X 100	Y 25
(P4)	X 75	Y 50
(P5)	X 50	Y 50
(P6)	X 25	Y 75
(P7)	X 100	Y 75
(P8)	X 100	Y 75







坐标变换

13

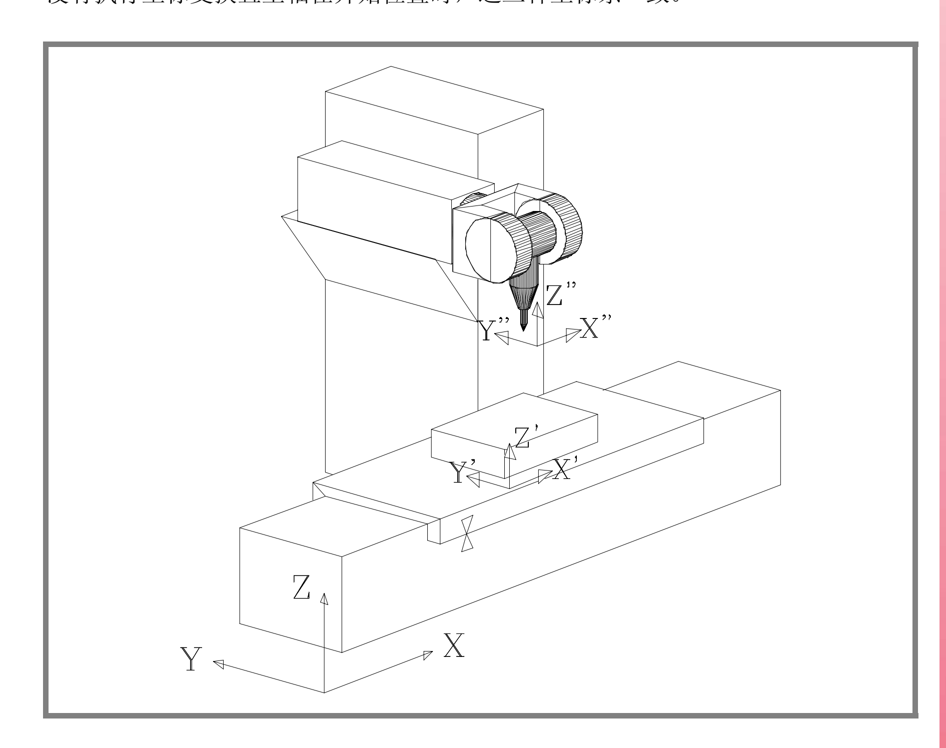
常规坐标变换的介绍可分为以下几方面进行：

- 运动学选择。指令 #KIN ID。
- 定义和选择加工坐标系 (包括平面)。指令 #CS。
- 定义和选择夹具坐标系。指令 #ACS 。
- RTCP (旋转刀具中心点) 变换。指令 #RTCP。
- 使刀具方向垂直工作平面 (平行于第三轴)。指令 #TOOL ORI 。
- 程序中的刀具长度补偿修正。指令 #TLC 。

为了清楚的进行对比，下例给出三种坐标系：

XYZ	机床坐标系
X' Y' Z'	工件坐标系
X" Y" Z"	刀具坐标系

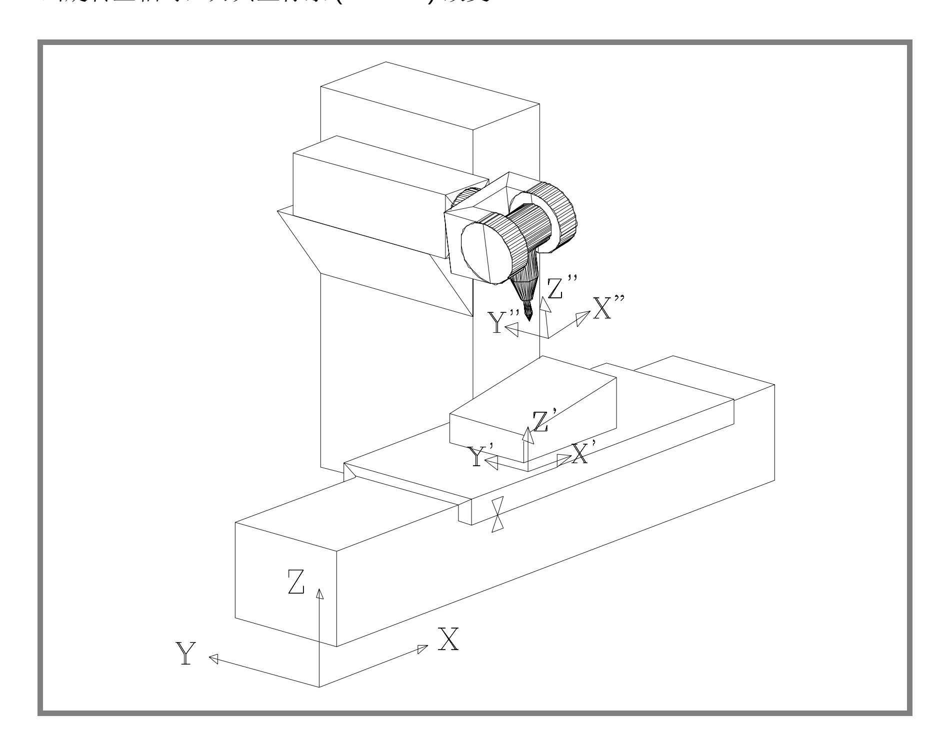
没有执行坐标变换且主轴在开始位置时，这三种坐标系一致。



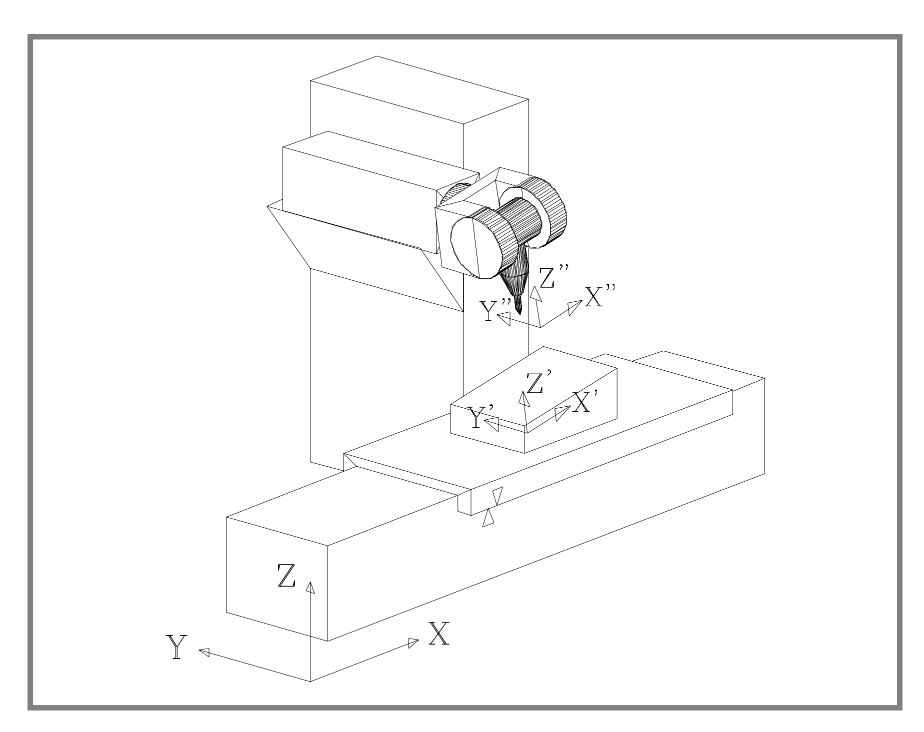




当旋转主轴时，刀具坐标系 (X" Y" Z") 改变 。



如果除此之外还选择新的加工坐标系 (#CS 指令) 或是夹具坐标系 (#ACS 指令)，工件坐标系也会改变 (X' Y' Z')。



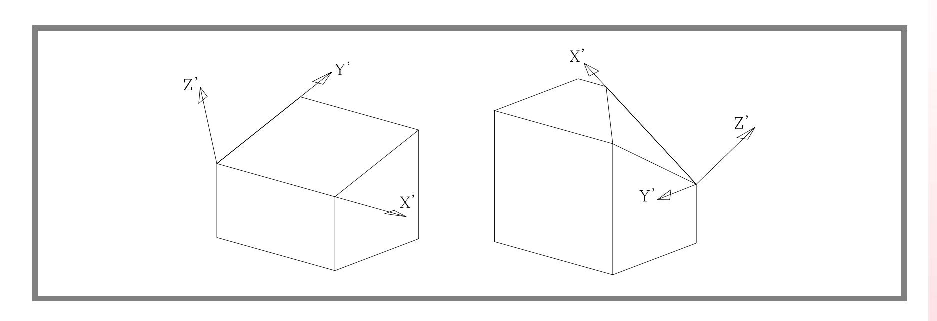


13.1 在斜面上的运动

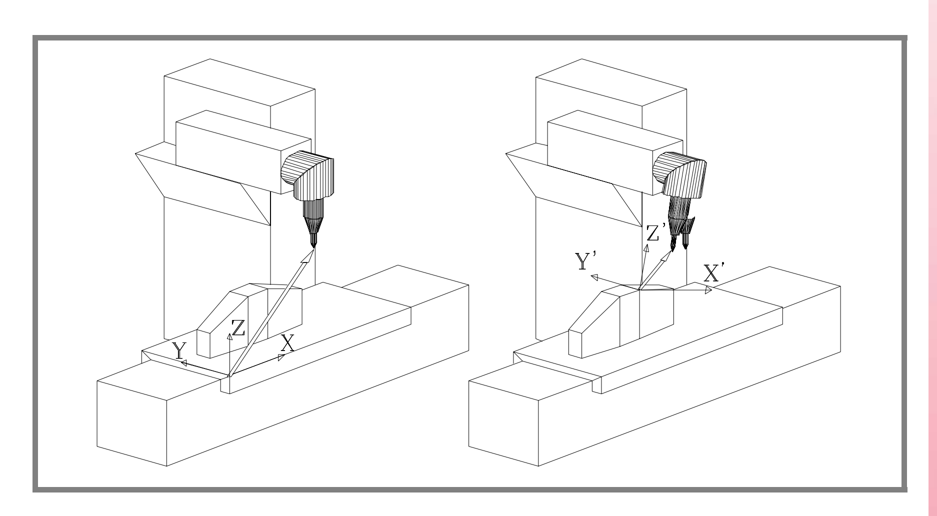
斜面可以是 XYZ 轴的坐标变换所形成的空间中的任意平面。

可以选择在空间中的任意平面上执行加工操作。

根据加工操作定义斜面，使用指令 #CS 和 #ACS，指令将在本章后面的小节介绍。

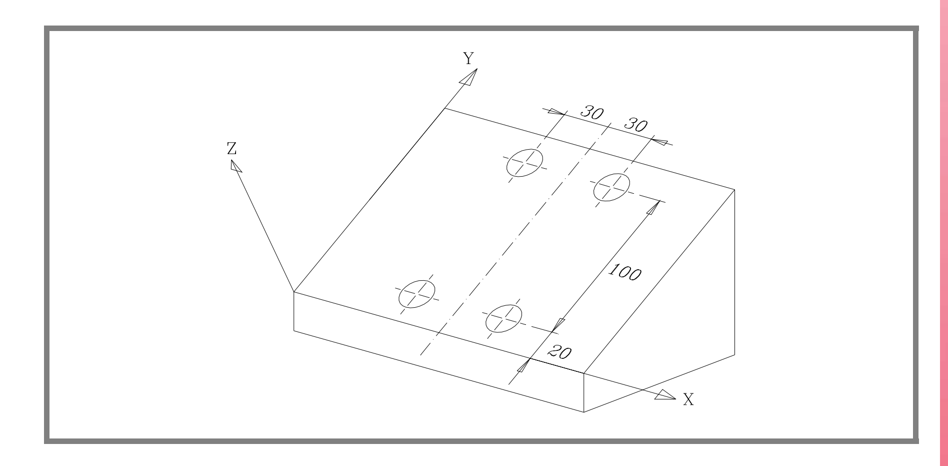


新坐标系 (右图) 是相对新工件零点的，假定刀具位于垂直新平面的方位。



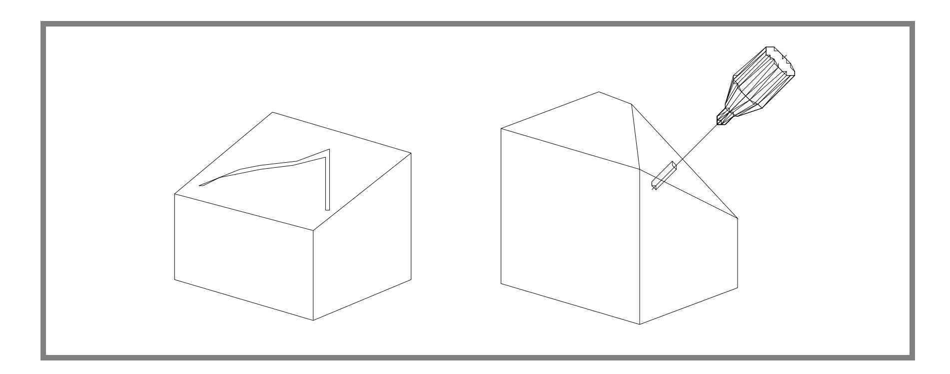
要把刀具放置到那个位置，可使用指令 #TOOL ORI ，或用表示位置的运动学相关变量，每个主轴箱的旋转轴必须占用。参见 340 页 “13.8 运动学相关参数”。

从现在起，编程以及 X, Y 轴的运动沿所选的平面执行， Z 轴运动则垂直于该平面。





使刀具和工件垂直于该平面，使用指令 #TOOL ORI，该指令在本章稍后介绍。





13.2 运动学选择 (#KIN ID)

OEM 可以为机床设置多达 6 个运动学。其中的每一个表示使用的主轴类型，特性和尺寸。

要执行变换，必须指定使用的运动学。通常，通过通用机床参数 KINIDOEM 定义缺省时使用的运动学编号。

如果定义几个运动学，可以使用指令 #KIN ID 通过工件程序激活想得到的那个。如果仅有一个且被设置成缺省运动学，不需要编写该指令。

激活特定运动学的格式：

```
#KIN ID [n]
```

n 运动学编号

在选择某个运动学后，必须激活功能 #RTCP, #TLC 和 #TOOL ORI。当 #RTCP 或 #TLC 有效时，不能改变运动学。

```
N50 #KIN ID[2]          ( 激活运动学 Nr 2)
N60 #RTCP ON            ( 用运动学 2 激活 RTCP)
...
N70 #RTCP OFF           ( 关掉 RTCP )
N80 M30
```




13.3 坐标系统 (#CS) (#ACS)

有两种不同类型的坐标系统，即加工坐标系和夹具坐标系。使用相关指令来操作每一种坐标系。

#CS	用指令 #CS , 可以定义、存储、激活、取消最多 5 个加工坐标系 。
#ACS	用指令 #ACS , 可以定义、存储、激活、取消最多 5 个加工坐标系 。用于补偿由于使用夹具而导致的工件平面倾斜。

两条指令使用相同的编程格式，并且可以单独或组合使用，在后面的小节会进行介绍。

几个 #ACS 和 #CS 坐标系可以组合在一起。当激活一个新的坐标系时，它被累加到当前坐标系上 。参见 330 页 "13.4 如何组合多个坐标系"。

推荐使用指令 #CS NEW 或 #ACS NEW 开始编程，以避免不想得到的平面。在中断程序或恢复执行后会发生这种情况。

坐标系和工件零点

坐标系原点是相对于当前工件零点的。当激活 #CS 或 #ACS 时，可能会在平面上预置新的工件零点。

如果取消一个倾斜平面，且没有定义新的平面，将恢复在激活倾斜平面之前定义的工件零点。可以根据需要，定义是否保存当前工件零点。

有时会发生这种情况，当激活一个早先保存过的 #CS 或 #ACS 时，平面的坐标原点可能不是想得到的那个。如果工件零点在定义和 #CS 或 #ACS 应用之间被修改了，这种情况发生。

两功能的注意事项

两个坐标系 (#CS 和 #ACS) 在复位 (Reset) 或 M30 之后保持有效。



关闭 CNC 时，将取消两个坐标系 (#CS 和 #ACS)，并删除早先保存的所有数据。

坐标系统操作

两条指令 (#CS and #ACS) 使用同一编程格式。两条指令使用的共同参数的含义的说明如下。带括号 [ ] 的表示必须编写。定在尖括号 "< >" 中间的参数表示可选用。

n	坐标系的编码 (1..5)。可以在任何时候定义和存储激活的最多 5 个不同的坐标系 。
MODE m	定义使用模式 (1..6)。
V1...V3	变换矢量的组成。
φ1...φ3	旋转角度。
<0/1>	平面阵列 (值 0/1)。仅在模式 3, 4 和 5 下。
<KEEP>	在变换中保持定义的工件零点。
<FIRST/SECOND>	轴定位。仅在模式 6 下。

定义方式

MODE 定义模式设置轴旋转到达想要的平面次序。在一些情形下，有两种得到平面的方案；执行选择，定义坐标系的哪根轴和平面保持成一排。

取消工件零点时保持工件零点

取消倾斜平面，如果没有定义其它的，恢复在激活倾斜平面前定义的工件零点。



命令 <KEEP> 可以用于保持当前工件零点。该命令仅允许在取消坐标系的指令中使用。

编程格式

- 定义和存储格式 :

```
#CS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
#ACS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
```

- 定义、存储和激活格式 :

```
#CS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
#ACS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
```

- 定义和激活 (没有存储) 格式 :

```
#CS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
#ACS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
```

  仅能定义它们中的一个；要定义另一个，必须撤销先前的一个。可以一直使用，直至被撤销，就如存储在存储器中的任意坐标系一样。

- 取消和删除所有当前 #CS o 或 #ACSF 并且定义和激活一个新的坐标系的格式 :

```
#CS NEW <KEEP> [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
#ACS NEW <KEEP> [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
```

- 取消和删除所有当前#CS o或#ACSF并且定义和激活(没有存储)一个新的坐标系的格式 :

```
#CS NEW <KEEP> [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
#ACS NEW <KEEP> [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>]
```

- 采用和存储当前坐标系作为 #CS 或 #ACS 的格式 :

```
#CS DEF ACT [n]
#ACS DEF ACT [n]
```

- 激活已经存储的一个坐标系的格式 :

```
#CS ON [n]
#ACS ON [n]
```

- 激活最近存储的坐标系的格式 :

```
#CS ON
#ACS ON
```

- 取消最近被激活的坐标系的格式 :

```
#CS OFF <KEEP>
#ACS OFF <KEEP>
```

- 取消所有被激活的 #CS or #ACS 的格式 :

```
#CS OFF ALL
#ACS OFF ALL
```

45º 主轴 (Huron 型)

Huron型主轴有两种方法使刀具垂直于新工作平面。对于这种类型的主轴，可以选择使用两种方法中的一种。参见 329 页 “13.3.7 主轴的操作 （Huron 型）”。



编程实例

```
#CS NEW [3] [MODE 1,2,15,5,2,3,4.5]
    ( 删除当前 CS)
    ( 定义和储存它为 ICS3)
#CS DEF [2] [MODE 1,P1,15,5,2,3,4.5]
    ( 定义和储存它为 CS2)
#CS DEF [5] [MODE 2,0,1,2,0,30,30]
    ( 定义和储存它为 CS5)
#CS ON
    ( 激活最近编写的 CS , CS5)
#CS OFF
    ( 取消 CS5)
#CS ON [3]
    ( 激活 CS3)
#CS DEF [2] [MODE 1,1,1.2,1.3,0,0,33]
    ( 重新定义储存过的 CS2， CS3 保持有效 )
M30
```

13.3.1 坐标系统定义模式 MODE 1

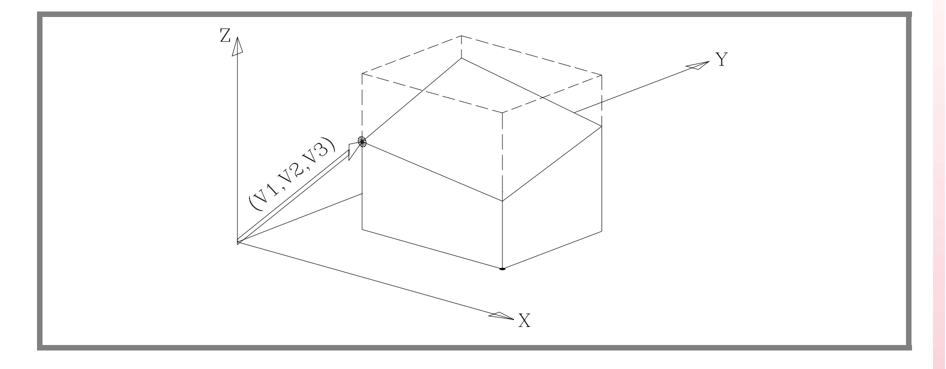
两条指令使用同一编程格式，可以组合或分开使用。

```
#CS DEF [n] [MODE 1, V1, V2, V3, φ1, φ2, φ3]
#ACS DEF [n] [MODE 1, V1, V2, V3, φ1, φ2, φ3]
```

定义倾斜平面，倾斜平面由先绕第一轴，然后是绕第二轴，最后是绕第三轴形成，旋转总量分别由 φ1, φ2, φ3 确定。

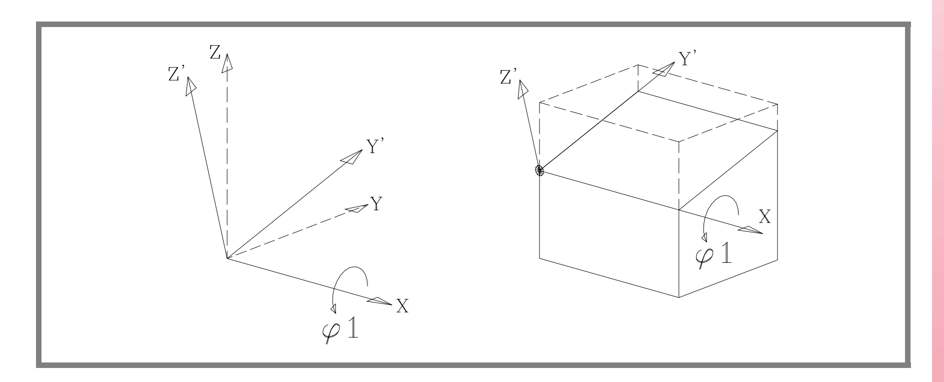
V1, V2, V3

相对于当前工件零点定义斜面的坐标原点。



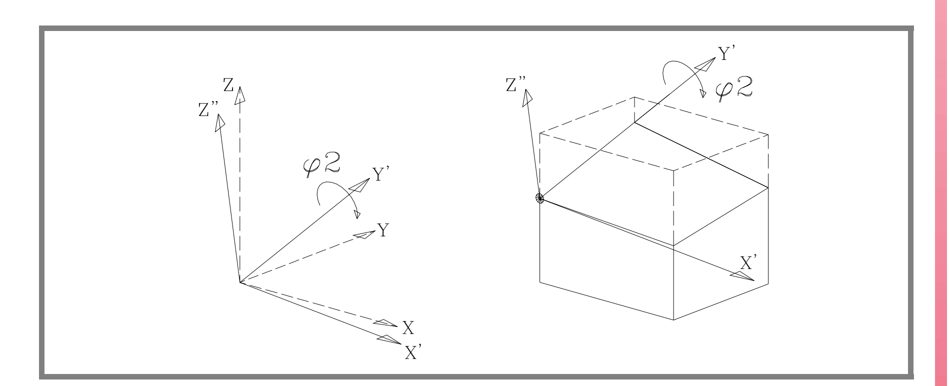
φ1, φ2, φ3

定义首先绕第一轴 (X) 旋转所得的倾斜平面，旋转角度由 φ1 给定。



在图中，由变换得到的新坐标系叫做 X Y' Z'，因为旋转了 Y, Z 轴。

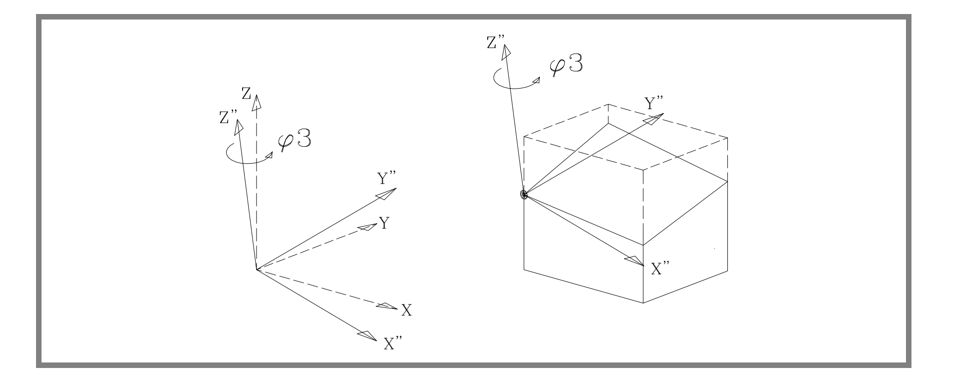
然后，绕第二轴 (Y') 旋转， φ2 表示旋转角度。



在图中，由变换得到的新坐标系叫做 X' Y' Z" ，因为旋转了 X, Z 轴。



最后，绕 Z" 轴旋转，φ3 表示旋转角度。





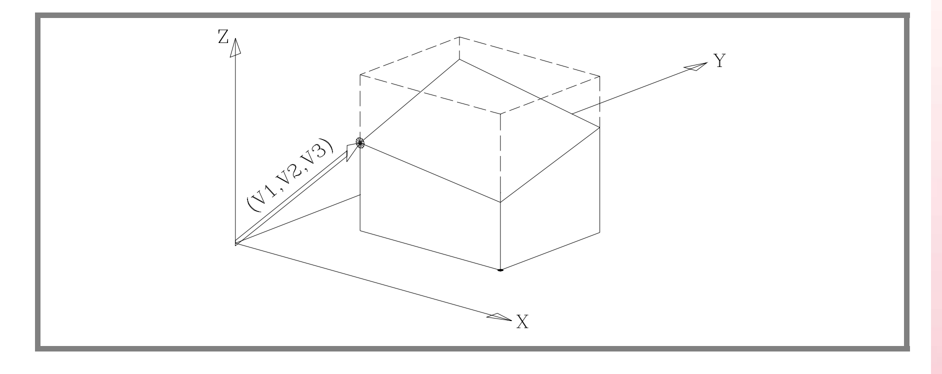
13.3.2 坐标系统定义模式 MODE 2

两条指令使用同一编程格式，可以组合或分开使用。

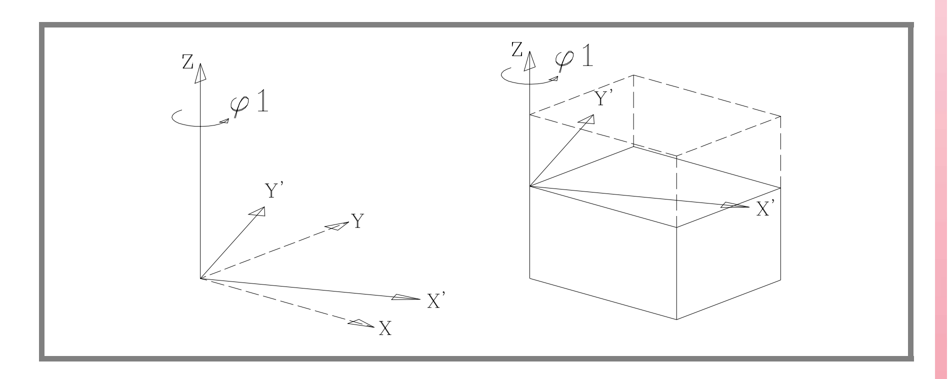
```
#CS DEF [n] [MODE 2, V1, V2, V3, φ1, φ2, φ3]
#ACS DEF [n] [MODE 2, V1, V2, V3, φ1, φ2, φ3]
```

用球坐标定义倾斜平面，倾斜平面由绕第三轴旋转，然后第二轴旋转，再次绕第三轴旋转而得到。旋转角度分别由 φ1, φ2, φ3 确定。

V1, V2, V3 相对于当前工件零点定义斜面的坐标原点。

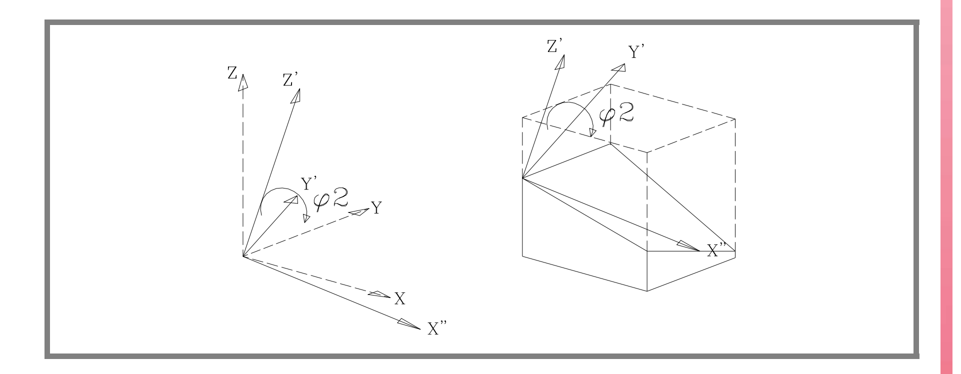


φ1, φ2, φ3 定义首先绕第三轴旋转所得的倾斜平面，旋转角度由 φ1 给定。



在图中，由变换得到的新坐标系叫做 X' Y' Z，因为旋转了 X, Y 轴。

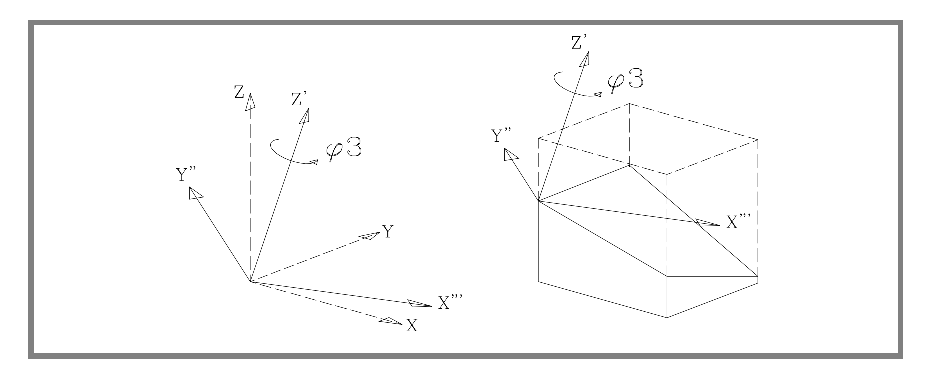
然后，必须绕 Y' 轴旋转 φ2 给出的旋转角度。



在图中，由变换得到的新坐标系叫做 X" Y' Z'，因为旋转了 X, Z 轴。



最后，绕 Z' 轴旋转 φ3 给定的角度。





13.3.3 坐标系定义模式 MODE 3

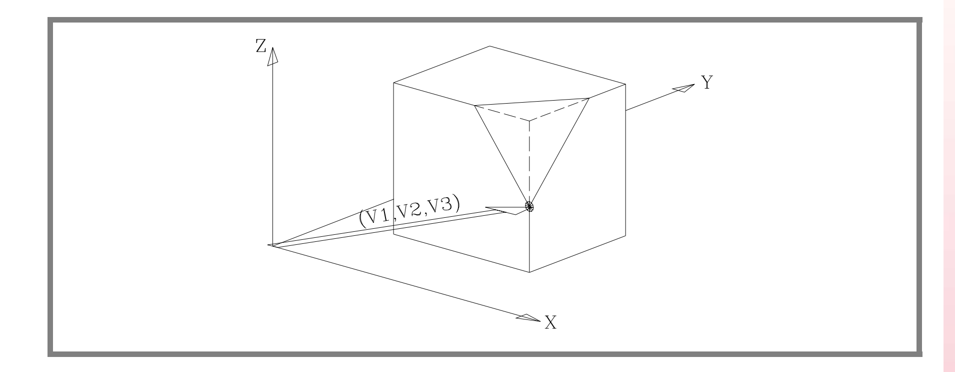
两条指令使用同一编程格式，可以组合或分开使用。

```
#CS DEF [n] [MODE 3, V1, V2, V3, φ1, φ2, φ3, <0/1>]
#ACS DEF [n] [MODE 3, V1, V2, V3, φ1, φ2, φ3, <0/1>]
```

使用相对于机床坐标系第一轴和第二轴 (X Y) 所成的角度定义倾斜平面。

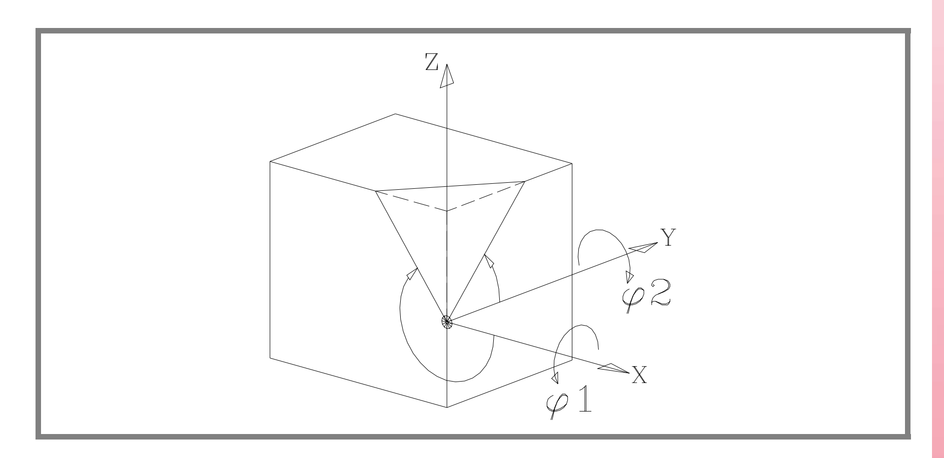
V1, V2, V3

相对于当前工件零点定义斜面的坐标原点。



φ1, φ2

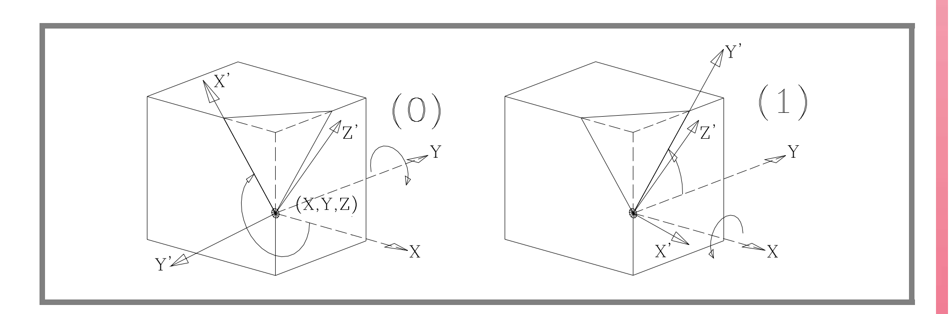
定义角度，倾斜平面和机床坐标系第一轴、第二轴所成的角度。



0/1

定义新平面的哪根轴和新平面的边缘成一行。

如果 <0> 是 X' 轴，如果 <1> 是 Y' 轴。如果没有编写，采用 <0>。



φ3

允许在新的笛卡尔平面 X' Y' 内定义和应用坐标旋转。



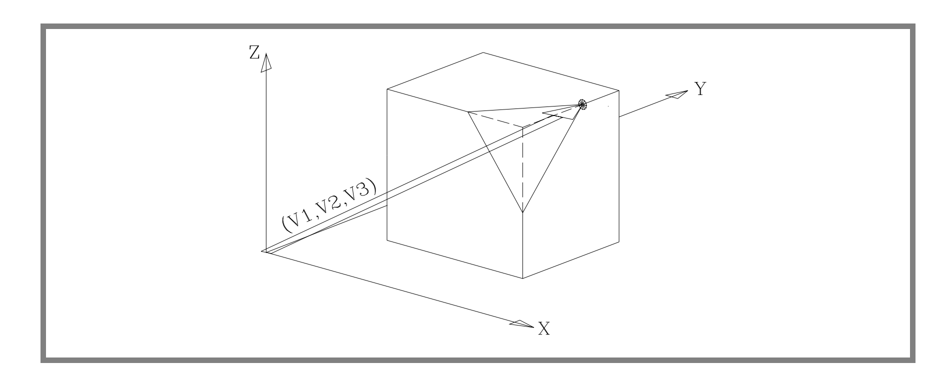
13.3.4 坐标系定义模式 MODE 4

两条指令使用同一编程格式，可以组合或分开使用。

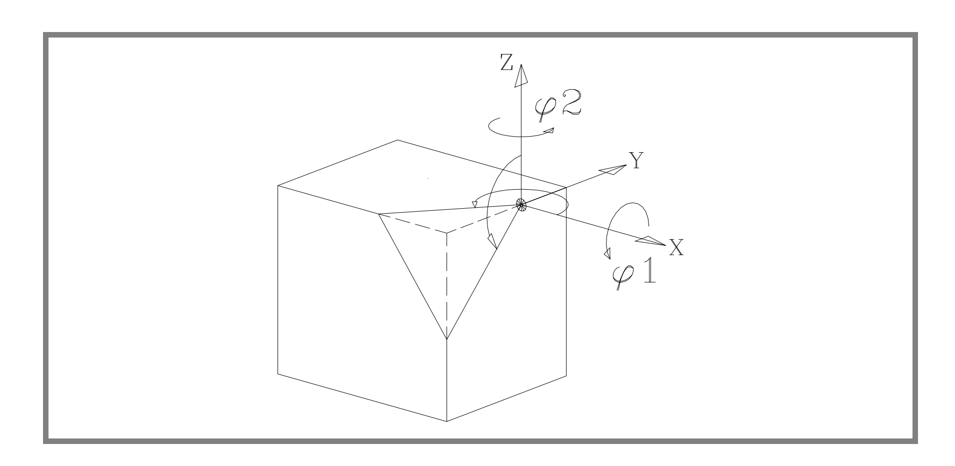
```
#CS DEF [n] [MODE 4, V1, V2, V3, φ1, φ2, φ3, <0/1>]
#ACS DEF [n] [MODE 4, V1, V2, V3, φ1, φ2, φ3, <0/1>]
```

使用相对于机床坐标系第一轴和第三轴 (X Z) 所成的角度定义倾斜平面。

V1, V2, V3 相对于当前工件零点定义斜面的坐标原点。

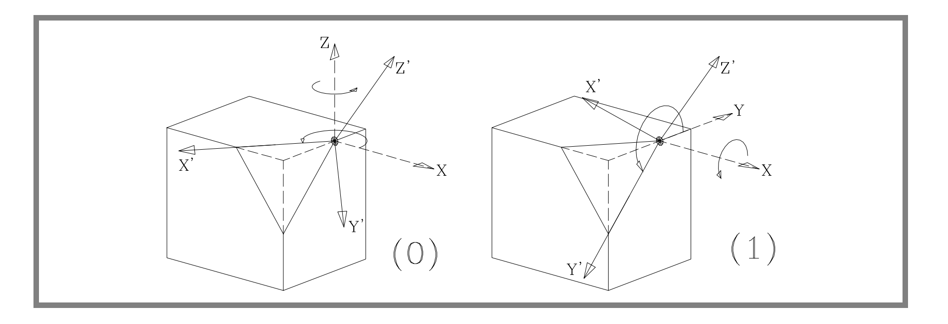


φ1, φ2 定义角度，倾斜平面和机床坐标系第一轴、第三轴所成的角度。



0/1 定义新平面的哪根轴和新平面 (X' Y') 边缘成一行。

如果 <0> 是 X' 轴，如果 <1> 是 Y' 轴。如果没有编写，采用 <0>。



φ3 允许在新的笛卡尔平面 X' Y' 内定义和应用坐标旋转。

13.3.5 坐标系定义模式 MODE5

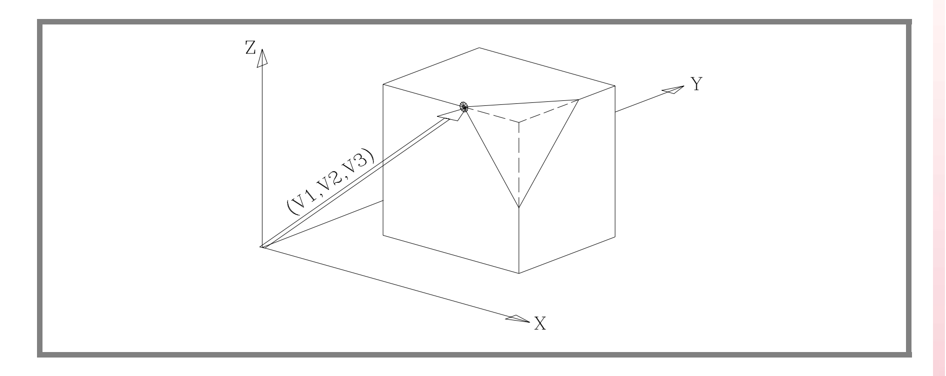
两条指令使用同一编程格式，可以组合或分开使用。

#CS DEF [n] [MODE 5, V1, V2, V3, φ1, φ2, φ3, <0/1>]

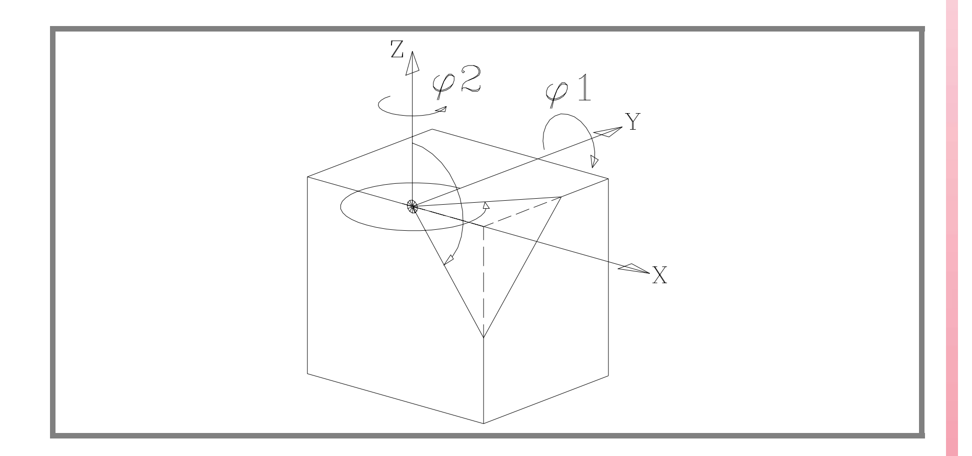
#ACS DEF [n] [MODE 5, V1, V2, V3, φ1, φ2, φ3, <0/1>]

使用相对于机床坐标系第二轴和第三轴 (Y Z) 所成的角度定义倾斜平面。

V1, V2, V3 相对于当前工件零点定义斜面的坐标原点。

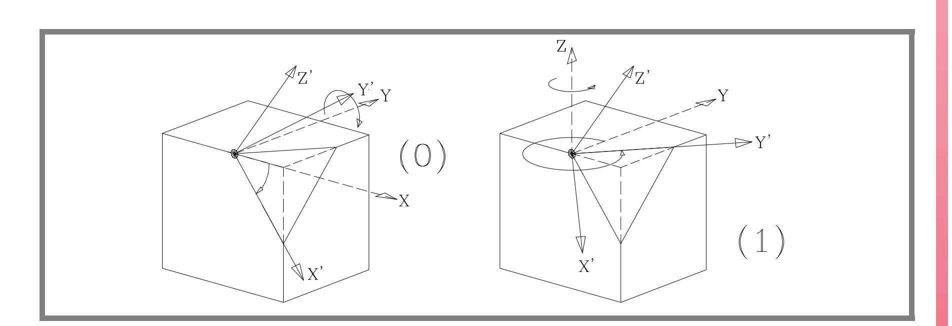


φ1, φ2 定义角度，倾斜平面和机床坐标系第二轴、第三轴所成的角度。



0/1 定义新平面的哪根轴和新平面 (X' Y') 边缘成一行。

如果 <0> 是 X' 轴，如果 <1> 是 Y' 轴。如果没有编写，采用 <0>。



φ3 允许在新的笛卡尔平面 X' Y' 内定义和应用坐标旋转。



13.3.6 坐标系统定义模式 MODE6



为了使用这种定义，在安装机床时，当刀具位置平行于机床的 Z 轴时，必须设置成主轴的静止位置。

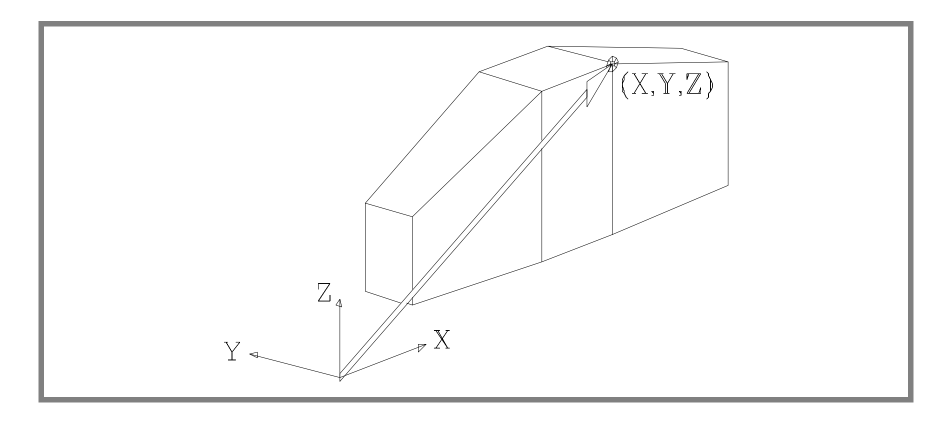
两条指令使用同一编程格式，可以组合或分开使用。

```
#CS DEF [n] [MODE 6, V1, V2, V3, φ1, <FIRST/SECOND>]
#ACS DEF [n] [MODE 6, V1, V2, V3, φ1, <FIRST/SECOND>]
```

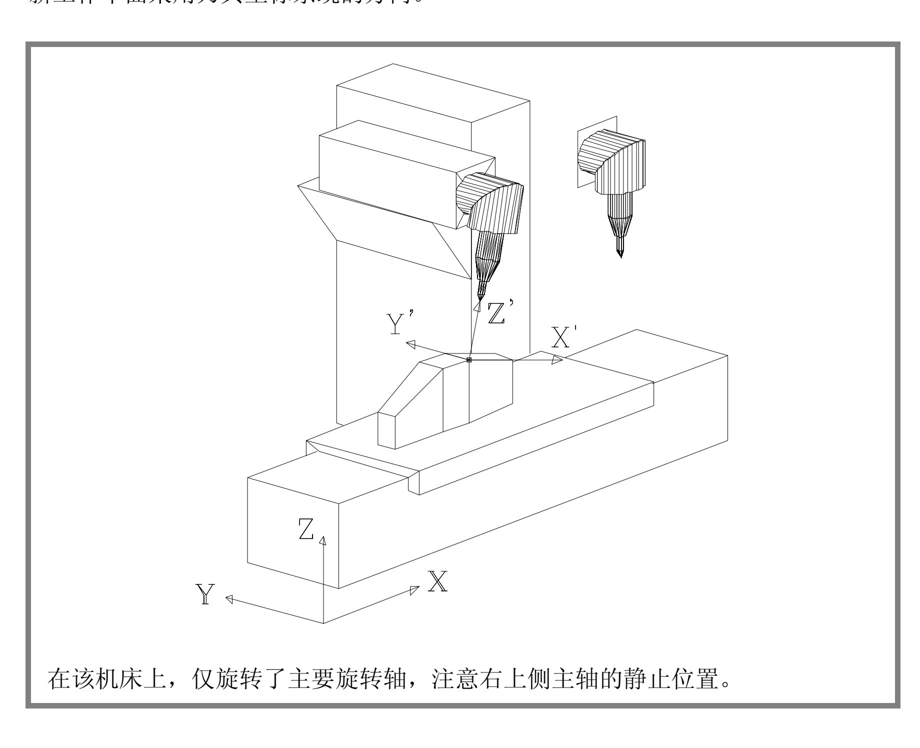
定义一个新工作平面 (倾斜平面) 垂直于刀具方向。

V1, V2, V3

相对于当前工件零点定义斜面的坐标原点。



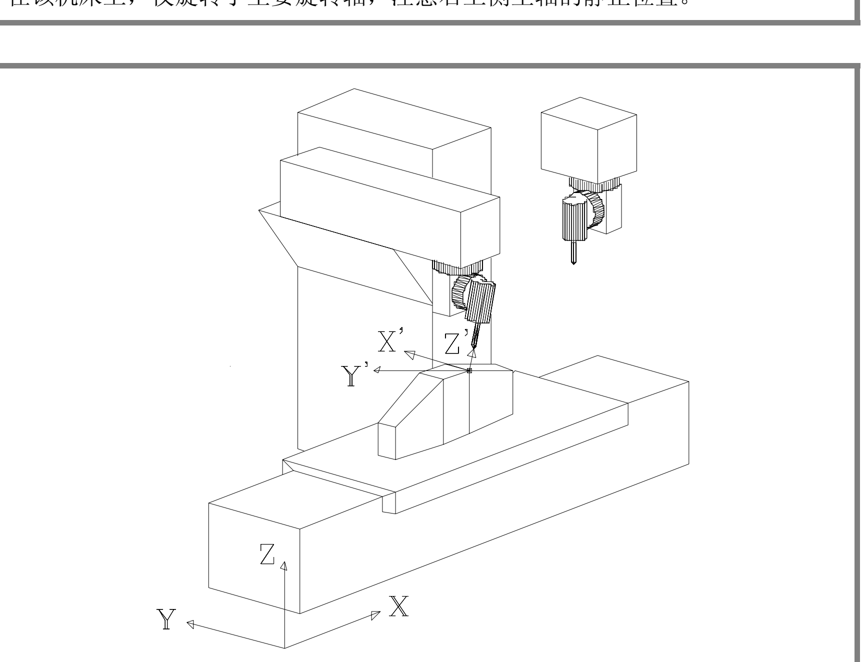
新工作平面采用刀具坐标系统的方向。



在该机床上，仅旋转了主要旋转轴，注意右上侧主轴的静止位置。



在该机床上，仅旋转了主要旋转轴，注意右上侧主轴的静止位置。



相反，在这台机床上，要实现相同的刀具定位，主要旋转轴和第二旋转轴都旋转了。注意右上侧的主轴静止位置。

主要轴旋转了 90º 度，所以，平面的 X' Y' 轴旋转了 90º。

φ1 允许在新的笛卡尔平面 X' Y' 内定义和应用坐标旋转。

如果在最后的机床上，想要如其他两种情形下那样定位 X', Y' 轴，必须按照如下方式编程：

```
#CS DEF [n] [MODE 6, V1, V2, V3, -90]
```




<FIRST/ SECOND>

当定义一个垂直于刀具的倾斜平面时，平面的第三轴完全使用刀具方位定义。另一方面，新平面的第一轴和第二轴的位置由主轴的类型确定；如果是 45º 主轴，很难预知。

根据编程选项，运转如下。

- 当编写命令 <FIRST> 时，倾斜平面的新第一轴的设计用机床第一轴确定方向。
- 当编写命令 <SECOND> 时，倾斜平面的新第二轴的设计用机床第二轴确定方向。
- 如果没有编写其中任何一个，不可能预先确定轴的方向，它将取决于主轴的类型。



13.3.7 45º 主轴的操作 (Huron 型)

Huron 型主轴对使刀具垂直于新工作平面有两种解决方案 。

- 第一种方案是相对于零点位置，使主旋转轴运动最少 （铰接式接头距滑枕最近或距刀具最近）。
- 第二种方案是相对于零点位置，使主旋转轴运动最多。

选择的方案将用于计算主轴的偏置，指令 #TOOL ORI 使刀具垂直于新平面。参见 332 页 “13.5 平面的刀具补偿 (#TOOL ORI)”。

选择一种解决方案来确定主轴方向

当定义一个新的坐标系时。可以定义使用两种方案中的哪一种。针对主轴类型，编写命令 <sol2> 应用第二种方案；反之，应用第一种方案。

```
#CS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#CS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#CS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#CS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#CS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]

#ACS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#ACS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#ACS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#ACS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
#ACS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, <0/1>, <SOL2>]
```

考虑每根轴占据的位置

每根旋转轴所占位置可以考虑下列变量。

- 针对第一种方案。

(V.)G.TOOLORIF1	主旋转轴位置。
(V.)G.TOOLORIS1	第二旋转轴位置。

- 针对第二种方案。

(V.)G.TOOLORIF2	主旋转轴位置。
(V.)G.TOOLORIS2	第二旋转轴位置。

每次数控系统更新这些变量时，通过指令 #CS 或 #ACS 选择新平面。



13.4 如何组合多个坐标系

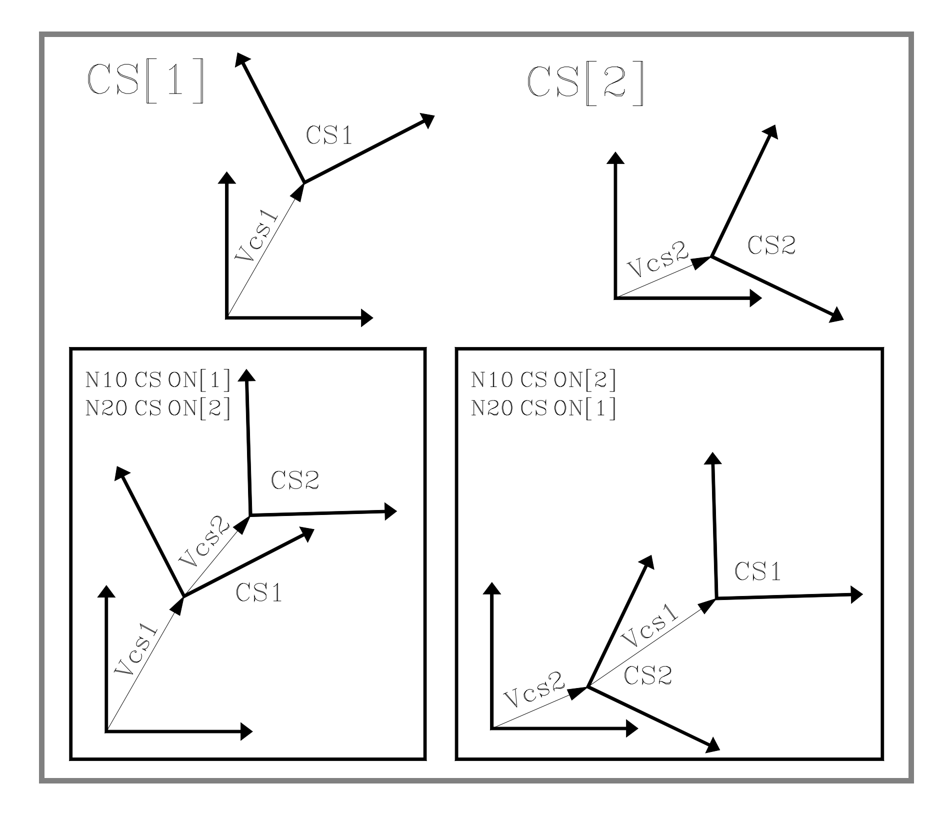
几个 #ACS 和 #CS 坐标系可以组合起来构建新的坐标系统。

例如，由工件上的一个夹具产生的倾斜 #ACS 坐标系可以和由加工工件的倾斜表面定义的 #CSF 坐标系组合在一起。

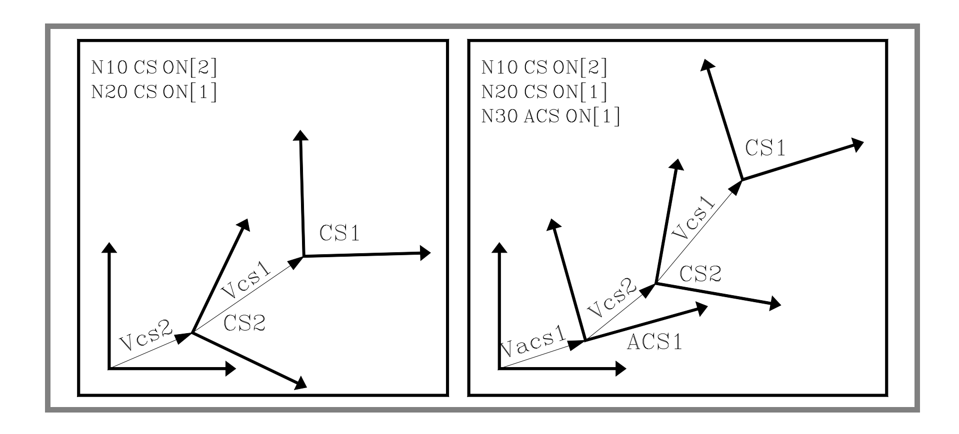
最多可以组合十个 #ACS 或 #CS 坐标系。数控系统工作如下：

1. 首先，按照编程的顺序检查 #ACS 并继而应用它们，产生一个 #ACS 变换。
2. 然后，按照编程的顺序检查 #CS 并继而应用它们，产生一个 #CS 变换。
3. 最后，在结果 #ACS 上应用结果 #CS 以获得新的坐标系。

组合的结果取决于被激活的顺序，如下图所示。



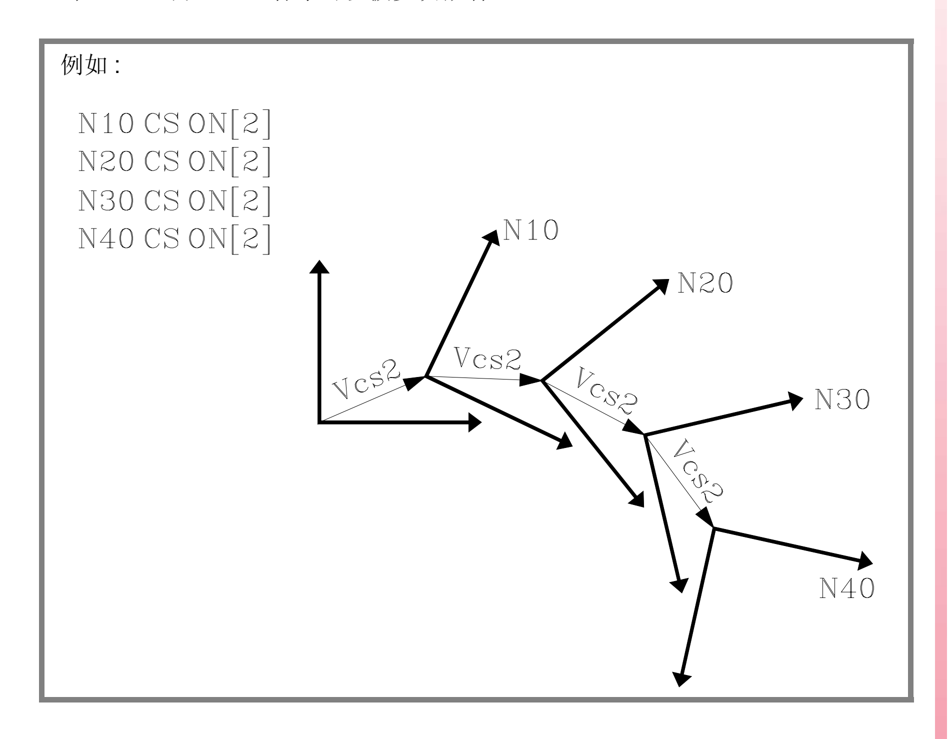
每次激活一个 #ACS 或 #CS，结果坐标系被重新计算，如下图中所示。



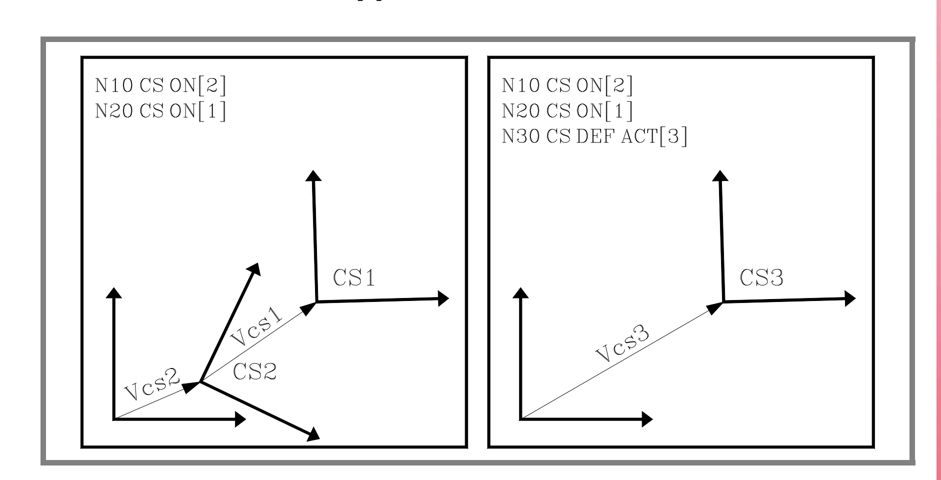
#ACS OFF 和 #CS OFF 指令分别取消最近被激活的 #ACS 或 #CS 。

`N100 #CS ON [1]`	(CS[1])
`N110 #ACS ON [2]`	(ACS[2] + CS[1])
`N120 #ACS ON [1]`	(ACS[2] + ACS[1] + CS[1])
`N130 #CS ON [2]`	(ACS[2] + ACS[1] + CS[1] + CS[2])
`N140 #ACS OFF`	(ACS[2] + CS[1] + CS[2])
`N140 #CS OFF`	(ACS[2] + CS[1])
`N150 #CS ON [3]`	(ACS[2] + CS[1] + CS[3])
`N160 #ACS OFF ALL`	(CS[1] + CS[3])
`N170 #CS OFF ALL`	
`M30`	

一个 #ACS 或 #CS 坐标系可以被多次激活。



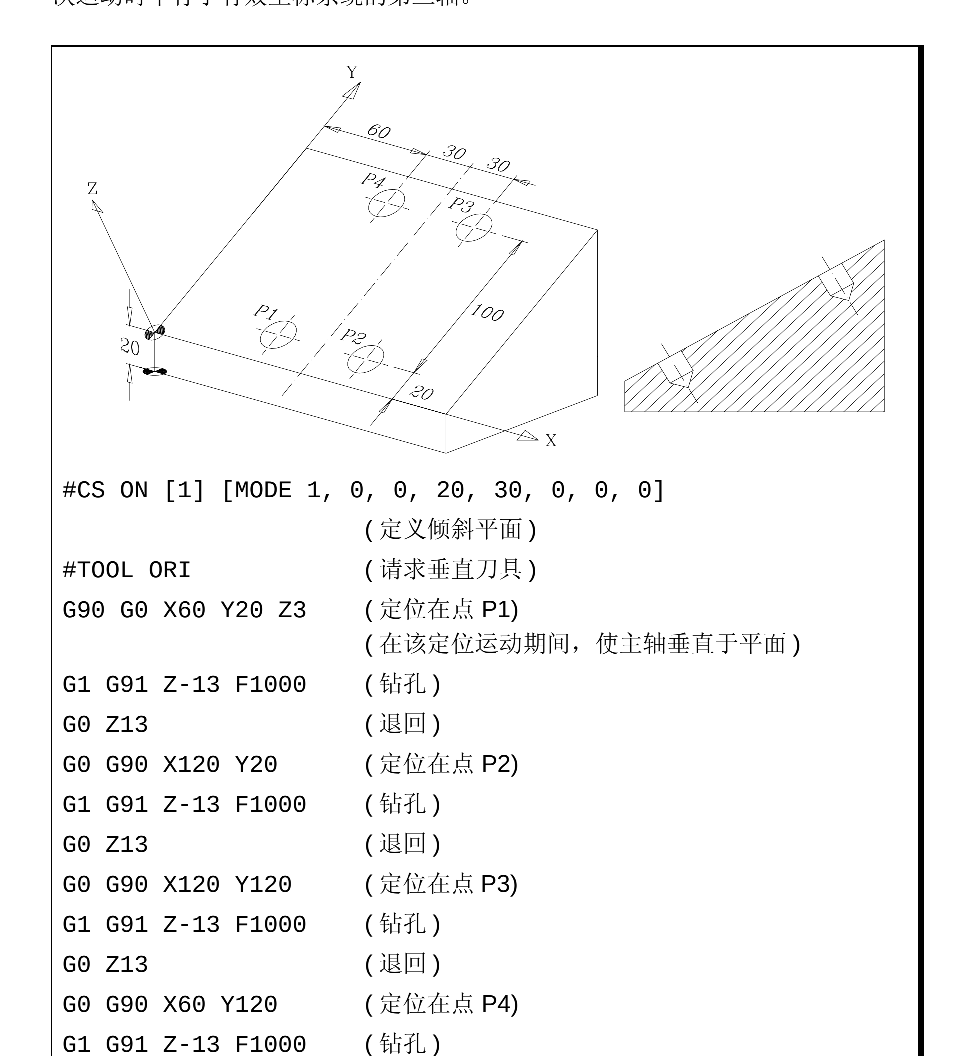
下图所示指令 #CS DEF ACT [n] 的例子采用并存储当前坐标系作为 #CS。



13.5 平面的刀具补偿 (#TOOL ORI)

指令 #TOOL ORI 用于放置刀具垂直于工作平面。

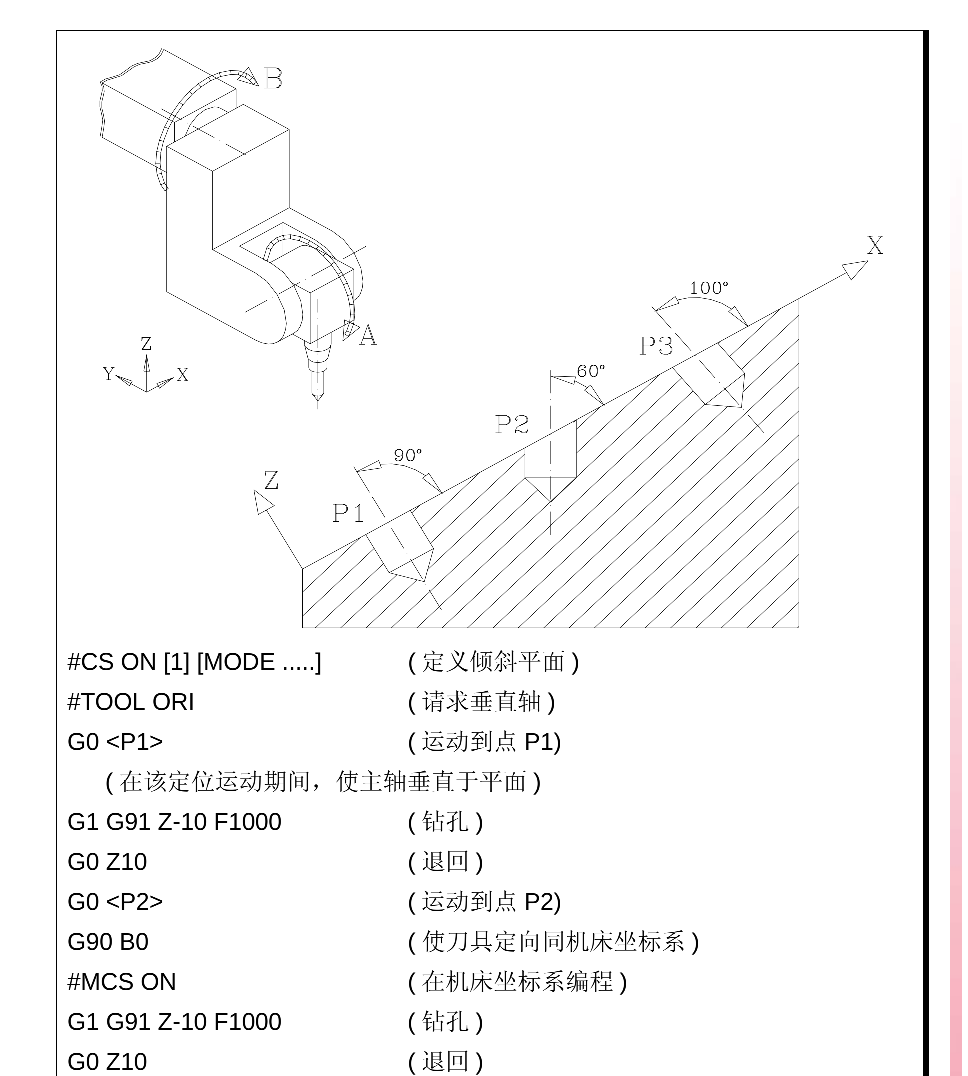
在执行指令 #TOOL ORI 之后，刀具定位在垂直于平面的位置，在紧接着编写的首次运动时平行于有效坐标系统的第三轴。




```
#CS ON [1] [MODE 1, 0, 0, 20, 30, 0, 0, 0]
                           (定义倾斜平面)
#TOOL ORI                  (请求垂直刀具)
G90 G0 X60 Y20 Z3          (定位在点 P1)
                           (在该定位运动期间，使主轴垂直于平面)
G1 G91 Z-13 F1000          (钻孔)
G0 Z13                     (退回)
G0 G90 X120 Y20            (定位在点 P2)
G1 G91 Z-13 F1000          (钻孔)
G0 Z13                     (退回)
G0 G90 X120 Y120           (定位在点 P3)
G1 G91 Z-13 F1000          (钻孔)
G0 Z13                     (退回)
G0 G90 X60 Y120            (定位在点 P4)
G1 G91 Z-13 F1000          (钻孔)
G0 Z13                     (退回)
M30
```




下例说明如何在相同的平面通过不同的倾斜完成钻三个孔：

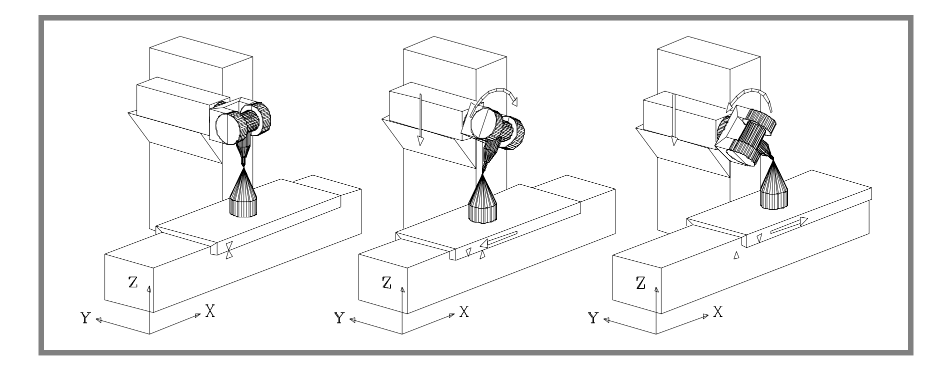



```
#CS ON [1] [MODE .....]        ( 定义倾斜平面 )
#TOOL ORI                      ( 请求垂直轴 )
G0 <P1>                        ( 运动到点 P1)
   ( 在该定位运动期间，使主轴垂直于平面 )
G1 G91 Z-10 F1000              ( 钻孔 )
G0 Z10                         ( 退回 )
G0 <P2>                        ( 运动到点 P2)
G90 B0                         ( 使刀具定向同机床坐标系 )
#MCS ON                        ( 在机床坐标系编程 )
G1 G91 Z-10 F1000              ( 钻孔 )
G0 Z10                         ( 退回 )
#MCS OFF                       ( 在机床坐标系编程结束，恢复工件平面坐标 )

G0 <P3>                        ( 运动到点 P3)
G90 B-100                      ( 使刀具定位在 100º)
#CS OFF
#CS ON [2] [MODE6 .....]       ( 定义垂直于刀具的倾斜平面 )
G1 G91 Z-10 F1000              ( 钻孔 )
G0 Z30                         ( 退回 )
#CS OFF
M30
```

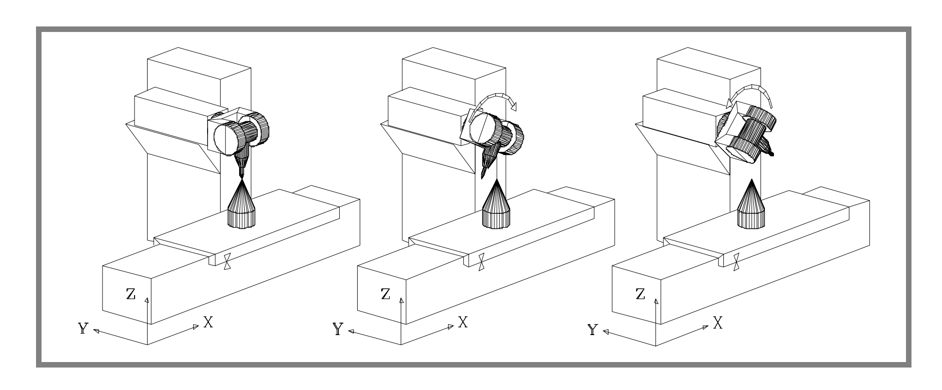
13.6 使用 RTCP (旋转式刀具中心点)

不修改刀尖在工件上所占的位置，改变刀具方向。RTCP 相当于瞬时的刀具补偿 。



显然，为了保持刀尖位置，数控系统必须一直移动多根轴。

下图说明当没有 RTCP 工作时，旋转主轴时发生的情况。

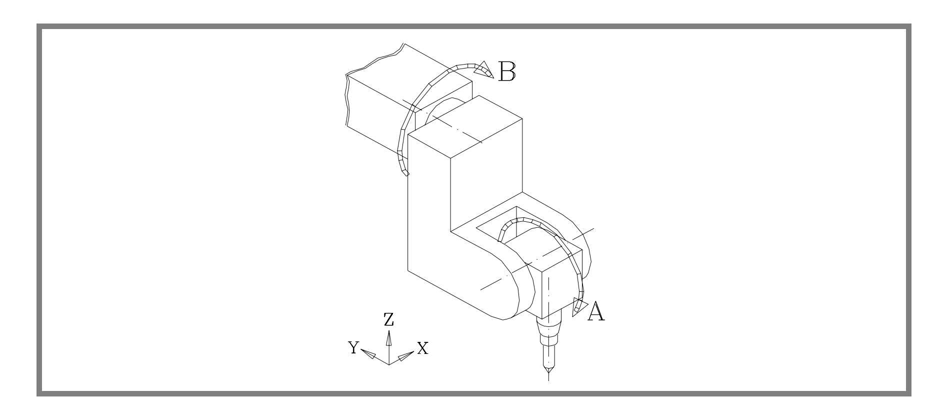


为应用 RTCP 变换，使用下列指令：

#RTCP ON	激活 RTCP 变换
#RTCP OFF	取消 RTCP 变换

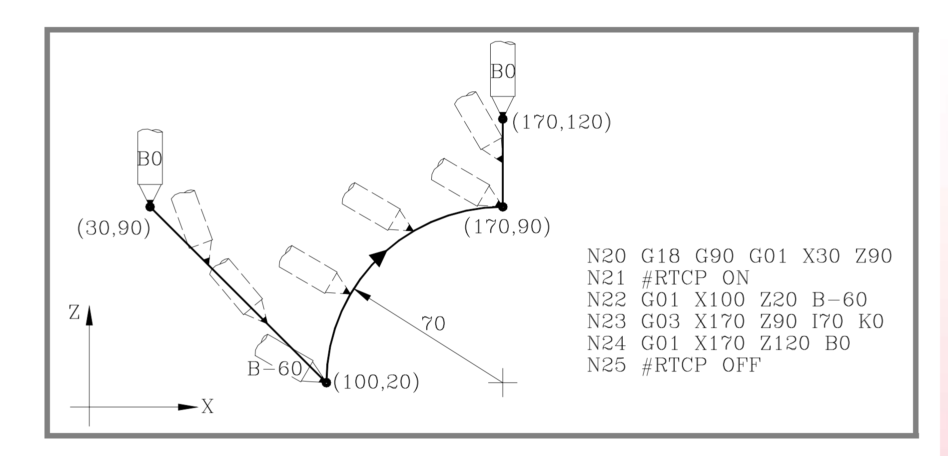
一旦 RTCP 变换有效，主轴定位可以与直线和圆弧插补结合起来。当 TLC 有效时，不能选择 RTCP 功能。

下例使用一双旋转矩形主轴头：





例 ·1·
保持刀具定位的圆弧插补



程序段 N20 选择 ZX 平面 (G18)，并且使刀具定位在起点 (30,90)。

程序段 N21 开启 RTCP。

程序段 N22 包含到点 (100,20) 的运动和刀具和从 0º 到 -60º 的定位。数控系统以刀具沿运动定位的方式，插补 X, Z 和 B 轴。

程序段 N23 进行圆弧插补到点 (170,90)，沿整个路径保持相同的刀具定位。

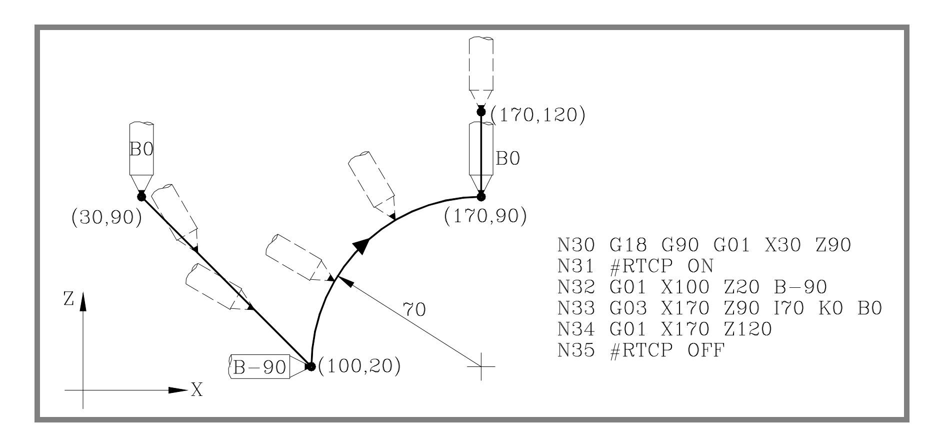
程序段 N24 包含到点 (170,120) 的运动和刀具从 -60º 到 0º 的定位。数控系统以刀具沿运动被定位的方式，插补 X, Z 和 B 轴。

程序段 N25 取消 RTCP 。



例 ·2·

刀具与路径垂直的圆弧插补



程序段 N30 选择平面 ZX (G18) 和使刀具位于起点 (30,90)。

程序段 N31 开启 RTCP 。

程序段 N32 包含到点 (100,20) 的运动和刀具 从 0º 到 -90º 的定位。数控系统以刀具沿运动被定位的方式，插补 X, Z 和 B 轴。 .

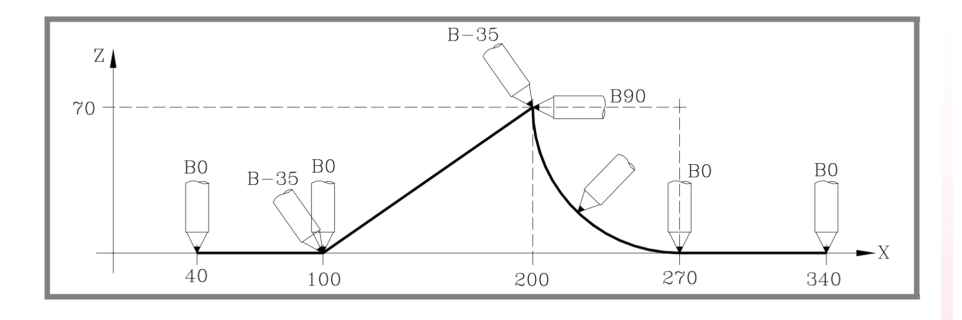
程序段 N33 包含圆弧插补到点 (170,90)，维持刀具一直垂直于路径 。

在起点定位到 -90º ， 在终点必须结束在 0º。数控系统插补 X, Z 和 B 轴，维持刀具一直垂直于路径 。

程序段 N34 移动刀具到点 (170,120)，保持 0º 方向。

程序段 N35 取消 RTCP。

例 ·3·
轮廓加工



`G18 G90`	选择 ZX 平面 (G18)
`#RTCP ON`	激活 RTCP 变换
`G01 X40 Z0 B0 F1000`	定位刀具在点 (40,0)，定向到 (0º)
`X100`	运动到点 (100,0)， 刀具定向到 (0º)
`B-35`	使刀具定向到 (-35º)
`X200 Z70`	运动到点 (200,70)，刀具定向到 (-35º)
`B90`	使刀具定向到 (90º)
`G02 X270 Z0 R70 B0`	圆弧插补到 (270,0)， 保持刀具垂直于路径
`G01 X340`	运动到点 (340.0)，刀具定向到 (0º)
`#RTCP OFF`	取消 RTCP 变换



13.6.1 RTCP 功能的注意事项

为了使用 RTCP 变换，必须定义 X, Y, Z 轴，它们必须形成三面体或成线性。X, Y 和 Z 可以是 GANTRY 轴。

RTCP 变换保持有效，甚至在执行 M02 或 M30，紧急情况 （Emergency）或复位 （Reset）和关闭数控系统之后。

当 RTCP 有效时，可以进行下列操作：

- 零点偏置 G54-G59, G159。
- 坐标系预置 (G92)。
- 在连续 / 增量方式下的手动和手轮运动。

如果 RTCP 变换有效时，不允许进行原点搜索 (G74) 。

当倾斜平面和 RTCP 变换同时工作时，推荐按照下列顺序编程：

```
#RTCP ON        ( 开启 RTCP )
#CS ON          ( 定义倾斜平面 )
#TOOL ORI       ( 刀具垂直平面 )
G               ( 开始加工 )
                ( 结束加工 )
#CS OFF         ( 取消倾斜平面 )
#RTCP OFF       ( 关闭 RTCP )
M30             ( 工件程序结束
```

应该首先启动 RTCP，因为它允许不修改刀尖位置而确定刀具方向 。



13.7 刀具长度补偿 (#TLC)

使用 CAD-CAM 生成程序，而且所用刀具没有相同的尺寸时，必须使用它。

#TLC 补偿长度差；而不能补偿半径差。

CAD-CAM 程序考虑刀具长度补偿，并生成刀具底部坐标。

使用功能 #TLC (刀具长度补偿) 时，数控系统补偿两刀具的实际和理论（计算）长度差。

可通过下列指令使用刀具长度补偿 (#TLC)：

#TLC ON [n]	开启 TLC。 n: 刀具长度差值 (实际 - 理论).
#TLC OFF	关闭 TLC。

当 RTCP 有效时，不能选择 TLC 功能。

```
N10 #TLC ON [1.5]      ( 开启 TLC， 刀具长度长 1.5mm)
N100 #TLC OFF          ( 关闭 TLC )
N200 #TLC ON [-2]      ( 开启 TLC， 刀具长度短 2mm)
N300 #TLC OFF          ( 关闭 TLC )
N200 M30
```




13.8 运动学相关参数

这些变量说明主轴头旋转轴所占的位置，和为了决定对定义平面刀具补偿位置它们必须占的位置。

当主轴不是全电动 （单一旋转或手动主轴）时，它们非常有用。

表示旋转轴的位置的参数。它们可以被读写 (R/W) ，单位为度。

(V.)G.POSROTF	主旋转轴位置。
(V.)G.POSROTS	第二旋转轴位置。

说明旋转轴的位置的参数必须挨次占用为了刀具对定义工作平面的补偿。它们是只读 (R) 并且以度为单位给出 。对旋转主轴有两种可能的解决方案：

使主旋转轴相对于零点位置的运动最短。

(V.)G.TOOLORIF1	主旋转轴定位，以垂直斜面定位。
(V.)G.TOOLORIS1	第二旋转轴定位，以垂直斜面定位。

这个包含主旋转轴的相对零点位置的最长运动。

(V.)G.TOOLORIF2	主旋转轴定位，以垂直斜面定位。
(V.)G.TOOLORIS2	第二旋转轴定位，以垂直斜面定位。

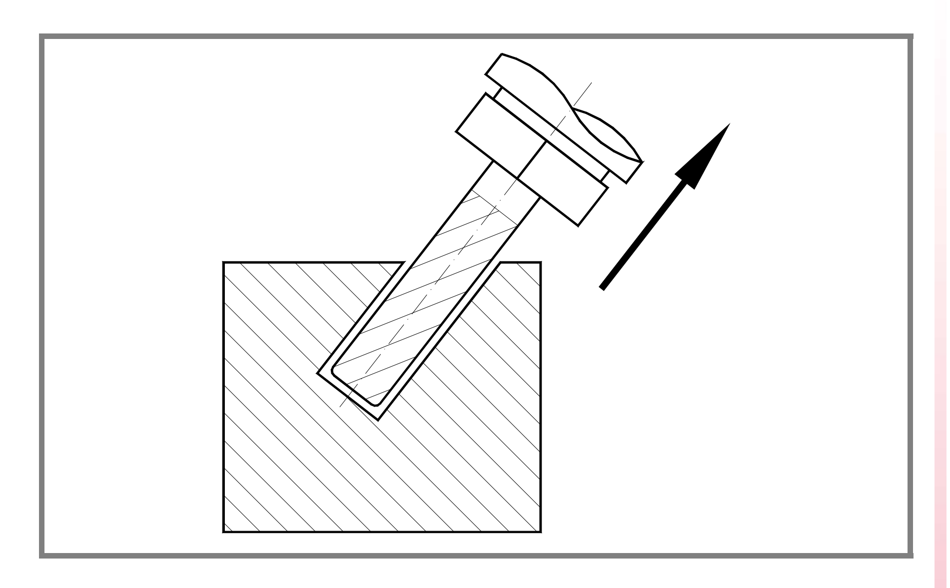
数控系统每当更新变量 (V.)G.TOOLORI* 时，使用指令 #CS 或 #ACS 选择新的平面。



13.9 丢失平面时如何退刀

当在运动学动作时，如果关闭后又开启数控系统，已选的工作平面丢失。

如果刀具在工件里面，可按如下步骤退刀：



使用指令 #KIN ID [n] 选择原来正在使用的运动学。

使用MODE6 坐标系定义，那么数控系统选择垂直于刀具方向的平面作为工作平面。

```
#CS ON [n] [MODE 6, 0, 0, 0, 0]
```

沿垂直轴移动刀具，直到刀具离开工件。

该运动可以在手动模式或通过编程执行，例如， G0 G91 Z20。




13.
坐标变换
丢失平面时如何退刀

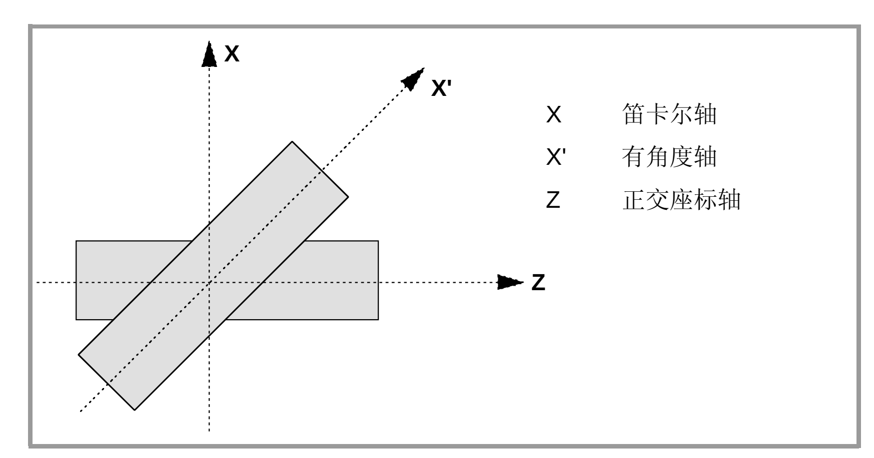

FAGOR

倾斜轴的角度变换

14

用倾斜轴的角度变换，有可能进行沿着轴的运动，轴不用垂直于另一个。运动在笛卡尔坐标系下编写，然后被转换到在实轴上的运动。

在某些机床上，轴被设置成笛卡尔状态，它们不相互垂直。一个典型的例子，车床的 X 轴由于硬性的理由不垂直于 Z 轴。



在笛卡尔系 (Z-X) 编程需要激活倾斜平面的角度变换，转换实 （非垂直）轴的运动 (Z-X')。这种方式，对X轴编写的运动被变换成Z-X'轴的运动；也就是沿Z轴和有角度 X' 轴的移动。

角度变换激活和取消

数控系统上电时采取没有变换；角度变换用指令 #ANGAX ON 通过部分编程被激活。几个角度变换可以同时有效。

角度变换用指令 #ANGAX OFF 通过部分编程被取消。随意地，一个变换可以被“冷冻”（暂时），通过在笛卡尔系编程去移动有角度轴 。

复位，关闭数控系统和在角度变换下执行 M30 的影响。

执行复位 （RESET）或 M30 之后一个倾斜轴的角度变换保持有效。关闭数控系统取消现行的角度变换。



CNC 8070

(SOFT V03.0X)

倾斜轴角度变换的注意事项。

编辑角度变换的轴必须满足下列要求 :

- 两个轴必须属于同一通道。
- 两个轴必须成线性的。
- 两个轴可以是组合 （从动）轴或是龙门式轴。

当角度变换有效时执行回复原位搜索是不可能的。

如果角度变换是有效的，坐标显示将是在笛卡尔系下的坐标。否则，将显示成实轴的坐标。



14.1 旋转角度变换激活和取消

激活角度变换

当角度变换在作用中，运动在笛卡尔系下被编写和进行，数控系统把它们转换到实轴的运动。在屏幕上显示的坐标将是在笛卡尔坐标系下的。

用指令 #ANGAX ON 激活角度变换，编程格式如下 :

```
#ANGAX ON [1,...,n]
```

1,...,n	被激活的角度变换编码。

在激活指令中必须至少有一个角度变换编码被编写 , 角度变换的数码由在机床参数表中它们被定义的顺序决定。

```
#ANGAX ON [1]
#ANGAX ON [5.7]
```

角度变换可以所有的同时被激活或者一个接一个的被激活。激活一个角度变换不取消先前的一个。

这个指令再一次激活 " 冻结 " (暂时地) 角度变换。 See "14.2 冻结角度变换 " on page 346.

取消角度变换

如果角度变换被取消，运动在实轴系被编写和执行。屏幕上显示的坐标将是实轴系下的。

用指令 #ANGAX OFF 取消角度变换，编程格式如下 :

```
#ANGAX OFF [1,...,n]
```

1,...,n	被激活的角度变换

如果没有角度变换被定义，通道的所有角度变换被取消。

```
#ANGAX OFF
#ANGAX OFF [1]
#ANGAX OFF [5.7]
```

在复位 （RESET）或 M30 之后倾斜轴的角度变换保持有效。关闭数控系统取消角度变换。



14.2 冻结角度变换

冻结角度变换是使运动沿有角度轴的一种特殊方法，但是在笛卡尔系下编写它。当手动方式下角度变换不能被 " 冻结 " (暂时) 。

用指令 #ANGAX SUSP 角度变换被 " 冻结 " (暂时)，编程格式如下 :

```
#ANGAX SUSP [1,...,n]
```

1,...,n　　被激活的角度变换数码。

如果没有编写任何角度变换编码 , 该通道的所有角度变换被 " 冻结 "(暂时)。角度变换的编码由它们在机床参数表里定义的顺序决定。

#ANGAX SUSP	冻结该通道的所有角度变换。
#ANGAX SUSP [1]	冻结角度变换 ·1·。
#ANGAX SUSP [5.7]	冻结角度变换 ·5· 和 ·7·。

在 " 冻结 " 角度变换之后编写运动

如果一个角度变换被 " 冻结 " (暂时), 仅有角度轴的坐标必须被编写在运动程序段。如果正交座标轴的坐标被编写，运动必须依照一般的角度变换被执行。

取消角度变换的冻结

角度变换的 " 冻结 " 在执行复位 （RESET）或 M30 之后被取消。

关于被冻结角度变换编写指令 #ANGAX ON 解除它的冻结。



14.3 角度轴结构校验

角度轴的结构数据可以直接在机床参数表中被校验或使用以下变量。

角度轴号码定义

(V.)MPK.NANG

从 PRG, PLC 和 INT 的只读

它返回在机床参数表中定义的角度变换的编码。

编辑角度变换的轴

括在括号里的有关角度变换的变量 *n.* 必须被编写。

(V.)MPK.ANGAXNA[n]

(V.)MPK.ORTGAXNA[n]

从 PRG, PLC and INT 的只读

第一个返回有角度轴的名字。第二个返回正交座标轴的名字。

角度变换的几何信息

括在括号里的有关角度变换的变量 *n.* 必须被编写。

(V.)MPK.ANGANTR[n]

从 PRG, PLC and INT 的只读

笛卡尔轴和相关联的有角度轴之间的夹角。当有角度轴是顺时针旋转 时是正角，相反逆时针时是负角。

(V.)MPK.OFFANGAX[n]

从 PRG, PLC and INT 的只读

有角度轴的原点的偏置。机床零点和倾斜轴的坐标系原点之间的距离 。







CNC 变量

15

15.1 变量描述的理解

PRG / PLC / INT – 访问变量

可从零件程序、MDI、PLC 及任何应用程序 （例如 FGUIM）中访问内部 CNC 变量。本章使用如下缩写来说明可访问这些变量的地方 ：

PRG	从零件程序或 MDI。
PLC	从 PLC。
INT	从任何应用程序 （接口）。例如：FGUIM。

每个变量必须指定是否是只读 (R) 或可读写 (R/W) 的。

LIN / ROT / CAB / ANA / SER – 与轴和驱动器相关的变量

与轴相关的变量指定轴的类型以及与变量相关的驱动器。

Lin	线性轴
Rot	旋转轴
Spd	主轴
Ana	模拟驱动器
Ser	Sercos 驱动器

使用 Sercos 驱动器时，它将指定该变量在位置模式 (P)、速度模式 (S) 或两种模式 (P/S) 下工作时是否有效。



EXEC – 在预备或执行期间访问变量

CNC 为了预先计算要跟随的路径，提前读取正在被执行的程序段之后的几个程序段（准备）。该预读过程称之为 “程序段准备”。

程序段准备期间将读取某些变量，而另外一些变量则在执行时才赋值。后者中断程序段准备。

(V.)G.PRGF　　使用 G94 编辑的进给率。在准备期间 （执行之前）赋值。

(V.)G.FREAL　　真实的 CNC 进给率。执行时赋值。

对于通过 PRG 访问的变量，“Exec”栏表示变量在程序段准备期间或执行期间是否被读写。

是　　执行时中断程序段准备过程。

否　　在准备期间。

从 PLC 或 INT 访问变量一定会中断程序段的准备过程。

中断程序段的准备可能会导致补偿路径与编程路径有出入，使用小片断工作时产生不期望的接合点，使用先行、急停轴运动时产生中断等。

当它正在被执行时，使用 #FLUSH 指令来强制为变量赋值。

Sync / Asyn – 通过 PLC 同步或异步访问变量

PLC 可以同步或异步访问 （读和写）变量。同步访问可以完成，但是异步访问需要几个 PLC 周期。

异步变量为：

- 当刀具既不处于激活状态又不在刀库中时，刀具变量将被异步读取。
- 无论刀具是否为激活状态，刀具变量将被异步写入。

关于如何访问异步变量的实例：

当它不在刀库中时，读取 9 号刀具偏置 1 的半径值。

```
<条件> AND NOT M11 = CNCRD (TM.TORT.[9][1], R11, M11)
```

在操作的开始，M11 标志被置 “1”，且它将保持该值一直到操作结束。

```
DFD M11 AND CPS R11 EQ 3 = ···
```

在计算数据之前，它等待查询结束。

关于如何访问同步变量的实例：

```
<条件> = CNCRD (G.FREAL, R12, M12)
CPS R12 GT 2000 = ···
```

不需要等待查询数据，因为同步变量可以马上完成。

```
<condition> = CNCWR (R13, PLC.TIMER, M13)
```

它通过寄存器 R13 中的值复位由 PLC 使能的时钟信号。



15.1.1 从 PLC 访问数字值

通过 PLC 访问可能带有小数的数字值时，必须记住按如下方式给出值：

坐标

如果以毫米为单位，以千分之十给出；如果以英寸为单位，以千分之一百给出。

对于 1 毫米	读数为 10000
对于 1 英寸	读数为 100000.
对于 1 度	读数为 10000.

轴的进给率

如果以毫米为单位，以千分之十给出；如果以英寸为单位，以千分之一百给出。

对于 1 毫米 / 分 .	读数为 10000.
对于 1 英寸 / 分	读数为 100000.

主轴转速

以千分之十给出。

使用 G97，对于 1 转 / 分 .　读数为 10000.

使用 G96，对于 1 米 / 分 .　读数为 10000.

使用 G96，对于 1 英尺 / 分 . 读数为 10000.

使用 G196 ，对于 1 转 / 分 .　读数为 10000.

使用 M19，对于 1 度 / 分 .　读数为 10000.

百分率

由变量决定是以十分之一或百分之一给出实际值。如果没有其它说明，将读取实际值。否则，它将指定是以十分之一 （x10）或者百分之一 （x100）读取该变量。

对于 1%	读数为 1.
对于 1%	(x10) 读数为 10.
对于 1%	(x100) 读数为 100.

时间

它们将以千分之一给出。

对于 1 秒	读数为 1000.

电压

与机床参数表格相关联的变量返回实际值 （以毫伏为单位）。对于其余的变量 （以伏为单位），则以千分之十给出。

对于 1 伏	读数为 10000



15.1.2 在单通道系统访问变量

变量名称

与变量相关的通用助记符按如下方式书写：

```
(V.){前缀}.{变量}
```

与变量相关的助记符以 (V.) 为开始。通过 PRG 访问它们时，使用这些字符（除去括号）。但是，从 INT 或 PLC 访问时不能使用它们。

助记符	PRG	PLC / INT
(V.)MPG.NAXIS	V.MPG.NAXIS	MPG.NAXIS

轴和主轴参数

轴和主轴变量用前缀 –A.– 来识别。如果是与主轴有关的变量，也可以通过前缀 –SP.– 来访问它们。

```
(V.)A.{变量}.{轴/主轴}
(V.)SP.{变量}.{主轴}
```

如果与主轴相关，以 –MPA.– 为前缀的机床参数变量也能够通过 –SP– 前缀访问。

```
(V.)MPA.{变量}.{轴/主轴}
(V.)SP.{变量}.{主轴}
```

在这些变量中，必须说明它们访问的轴或主轴。可以通过轴名或逻辑号访问轴；可以通过主轴的名字、逻辑号或主轴系统的索引或通道索引访问主轴。

识别轴和主轴

在以 –A.– 和 –MPA.– 为前缀的变量中，通过它们的逻辑号来识别轴和主轴。

- 对于轴，逻辑号设置指令 AXISNAME。
- 对于主轴，由 NAXIS + 指令 SPDLNAME 之和给出逻辑号。

在以 –SP.– 为前缀的变量中，根据指令 SPDLNAME，利用它们在系统中的索引来识别主轴。

主控主轴变量

它们是用于访问主控主轴数据的专用变量，不用知道它的名字及编号。它们用来显示数据和编程循环。

利用 –SP.– 前缀来识别这些变量，但是不指定主轴。

(V.)SP.{var}　　　　主控主轴变量

助记符	轴	主轴	主要主轴
(V.)A.POS.Xn	V.A.POS.X V.A.POS.1	V.A.POS.S V.SP.POS.S V.A.POS.6 V.SP.POS.2	V.SP.POS
(V.)MPA.AXISTYPE.Xn	V.MPA.AXISTYPE.X V.MPA.AXISTYPE.1	V.MPA.AXISTYPE.S V.SP.AXISTYPE.S V.MPA.AXISTYPE.6 V.SP.AXISTYPE.2	V.SP.AXISTYPE



15.1.3 访问单通道系统的变量

变量名

与变量相关的通用助记符按如下方式书写：

(V.)[通道].{ 前缀 }.{ 变量 }

与每个变量相关的助记符以 (V.) 为开始。通过 PRG 访问时，使用这些字符（除去括号）。但是，从 INT 或 PLC 访问时不能使用它们。

对于这些变量，必须指定它们所属的通道（第一个通道为编号 1，“0”为无效编号），必须编写括号。

助记符	PRG	PLC / INT
(V.)[n].G.FREAL	V.[1].G.FREAL	[1].G.FREAL

通道编程为可选项。如果没有指定通道，它将按照如下顺序默认通道：

PRG 正在执行的通道

PLC 第一通道或主通道

INT 有效通道

轴和主轴参数

轴和主轴变量用前缀 –A.– 来识别。与主轴相关的变量也可以使用前缀 –SP.– 来访问。

(V.)[n].A.{ 变量 }.{ 轴 / 主轴 }

(V.)[n].SP.{ 变量 }.{ 主轴 }

如果与主轴相关，以 –MPA.– 为前缀的机床参数变量也能够通过 –SP– 前缀访问。

(V.)MPA.{ 变量 }.{ 轴 / 主轴 }

(V.)SP.{ 变量 }.{ 主轴 }

在这些变量中，必须说明它们访问的轴或主轴。可以通过轴名或逻辑号访问轴；可以通过主轴的名字、逻辑号或主轴系统的索引或通道索引访问主轴。



识别轴和主轴

在以 –A.– 和 –MPA.– 为前缀的变量中，利用它们的逻辑号来识别轴和主轴。

- 对于轴，逻辑号设置指令 AXISNAME。
- 对于主轴，由 NAXIS + 指令 SPDLNAME 之和给出逻辑号。

在以 –SP.– 为前缀的变量中，利用它们的通道索引或系统索引来识别主轴。

- 当从编程接口或 PLC 读取时，可根据指令 SPDLNAME，利用它的系统索引来识别主轴。
- 当从编程接口（INT）读取时，根据指令 CHSPDLNAME，利用它的通道索引来识别主轴。

访问轴和主轴的通用变量

通过名字访问变量

通过名字访问轴或主轴时，对其所在的通道进行编程并非决定因素，因而，在这种情况下对它们进行编程是无关紧要的。当对通道进行编程时，如果轴或主轴不在其中，那么将忽略该程序。

(V.)A.{var}.X	使用该名字的轴变量。
(V.)A.{var}.S	使用该名字的主轴变量。
(V.)SP.{var}.S2	使用该名字的主轴变量。

通过逻辑号访问变量

根据是否编写了通道号，是通过 PRG、PLC 还是 INT 中访问，助记符有不同的含义。

在没有指定通道编号时，通过 PRG 或 PLC 中访问。

V.A.{var}.m	有逻辑号 *m* 的轴或主轴变量。
V.SP.{var}.m	系统中有 *m* 索引的主轴变量。

在没有指定通道编号时，从 INT 中访问。

A.{var}.m	有效通道中有 *m* 索引的轴变量。
SP.{var}.m	有效通道中有 *m* 索引的主轴变量。

在指定通道编号时，通过 PRG、PLC 或 INT 中访问。

(V.)[1].A.{var}.m	在通道中有 *m* 索引的轴变量。 (n=1 对应通道的第一轴)
(V.)[2].SP.{var}.m	在通道中有 *m* 索引的主轴变量。 (n=1 对应通道的第一主轴)

当指定通道编号时，不能够使用 –A.– 前缀访问主轴变量。

访问专用主轴变量

通过它们的名字来访问变量

对它们的访问和运行如同轴或主轴变量一样是相同的。

通过它们的逻辑号来访问变量

根据是否编辑通道编号，是否通过 PRG、PLC 或 INT 访问，助记符有不同的含义。

在没有指定通道编号时，通过 PRG 或 PLC 访问如同轴和主轴变量一样。

`(V.)A.{var}.m`	有逻辑号 *m* 的主轴变量。
`(V.)SP.{var}.m`	在系统中有 *m* 索引的主轴变量。

在没有指定通道编号时，通过 INT 访问。不能够从接口处 使用 –A.– 前缀来访问主轴变量。

`V.SP.{var}.m`	在有效通道中有 *m* 索引的主轴变量。

在指定通道编号时，通过 PRG、PLC 或 INT 访问。不能够使用 –A.– 前缀来访问主轴变量。

`(V.)[n].SP.{var}.m`	在 *n* 通道中有 *m* 索引的主轴变量。

主要主轴变量

它们是专用变量，在不知道名字、逻辑号或索引的情况下，可用它们来访问每个通道的主要主轴的数据。它们意味着显示数据和编程循环。

可用前缀来识别这些变量，但是不指定编号及主轴的名字。

`(V.)[n].SP.{var}`	通道主要主轴 *n* 的变量。

如果未对通道进行编程，将采用缺省通道，即：

PRG	正在执行的通道。
PLC	第一通道或主通道。
INT	有效通道。



15.2 与通用机床参数相关的变量

这些变量为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

- 使用轴号替代字母 “x”。
- 使用带括号的数字替代字母 “i” 和 “m”。

(V.)MPG.AXISNAMEx	V.MPG.AXISNAME2	V.MPG.AXISNAME3
(V.)MPG.MASTERAXIS[i]	V.MPG.MASTERAXIS[1]	V.MPG.MASTERAXIS[2]

通道配置		PRG	PLC	INT
(V.)MPG.NCHANNEL	CNC 通道数量	R	R	R

轴配置		PRG	PLC	INT
(V.)MPG.NAXIS	由 CNC 控制的轴的数量	R	R	R
(V.)MPG.AXISNAMEx	"n" 逻辑轴的命名	—	—	R
(V.)MPG.TMASTERAXIS[i]	级联式 [i]. 主控轴逻辑号	R	R	R
(V.)MPG.TSLAVEAXIS[i]	级联式 [i]. 从动轴逻辑号	R	R	R
(V.)MPG.TORQDIST[i]	级联式 [i]. 扭矩分布	R	R	R
(V.)MPG.PRELOAD[i]	级联式 [i]. 预载	R	R	R
(V.)MPG.PRELFITI[i]	级联式 [i]. 施加预载的时间	R	R	R
(V.)MPG.TPROGAIN[i]	级联式 [i]. 比例增益	R	R	R
(V.)MPG.TINTIME[i]	级联式 [i]. 积分增益	R	R	R
(V.)MPG.TCOMPLIM[i]	级联式 [i]. 补偿极限	R	R	R
(V.)MPG.MASTERAXIS[i]	龙门式 [i]. 主控轴逻辑号	R	R	R
(V.)MPG.SLAVEAXIS[i]	龙门式 [i]. 从动轴逻辑号	R	R	R
(V.)MPG.WARNCOUPE[i]	龙门式 [i]. 发布警告的最大差值	R	R	R
(V.)MPG.MAXCOUPE[i]	龙门式 [i]. 允许的最大差值	R	R	R
(V.)MPG.DIFFCOMP[i]	龙门式 [i]. 误差差值补偿 "0" = 否 "1"= 是	R	R	R

TORQDIST, PRELOAD, TPROGAIN 和 *TCOMPLIM* *的PLC读数*以百分之一(x100)给出。参考313页 *“从PLC 访问数字值”*。

主轴配置		PRG	PLC	INT
(V.)MPG.NSPDL	由 CNC 控制的轴的数量	R	R	R
(V.)MPG.SPDLNAMEx	"x" 主轴的命名	—	—	R

时间设置		PRG	PLC	INT
(V.)MPG.LOOPTIME	循环时间	R	R	R
(V.)MPG.PRGFREQ	PRG 模块的频率 （循环中）	R	R	R

CAN 和 SERCOS 总线设置		PRG	PLC	INT
(V.)MPG.SERBRATE	Sercos 传送速率 "0" = 4 兆比特每秒 "1" = 2 兆比特每秒	R	R	R
(V.)MPG.SERPOWSE	Sercos 光强度	R	R	R
(V.)MPG.CANLENGTH	CAN 总线电缆长度 (单位：米) "0" = 最大 20 "1" = 最大 30 "2" = 最大 40 "3" = 最大 50 "4" = 最大 60 "5" = 最大 70 "6" = 最大 80 "7" = 最大 90 "8" = 最大 100 "9" >100	R	R	R
(V.)MPG.CANMODE	CAN 总线类型 "0" = CANfagor "1" = CANopen	R	R	R

缺省条件		PRG	PLC	INT
(V.)MPG.INCHES	缺省工作单位 "0" = 毫米 "1" = 英寸	R	R	R

相关算术参数		PRG	PLC	INT
(V.)MPG.MAXLOCP	最大局部算术参数	R	R	R
(V.)MPG.MINLOCP	最小局部算术参数	R	R	R
(V.)MPG.MAXGLBP	最大全局算术参数	R	R	R
(V.)MPG.MINGLBP	最小全局算术参数	R	R	R
(V.)MPG.ROPARMAX	最大全局只读算术参数	R	R	R
(V.)MPG.ROPARMIN	最小全局只读算术参数	R	R	R
(V.)MPG.MAXCOMP	最大通用算术参数	R	R	R
(V.)MPG.MINCOMP	最小通用算术参数	R	R	R

交叉补偿表格		PRG	PLC	INT
(V.)MPG.MOVAXIS[m]	表格 [m]. 主控轴	R	R	R
(V.)MPG.COMPAXIS[m]	表格 [m]. 被补偿轴	R	R	R
(V.)MPG.NPCROSS[m]	表格 [m]. 点的数量	R	R	R
(V.)MPG.TYPCROSS[m]	表格 [m]. 补偿类型 "0" = 实际坐标 "1" = 理论坐标	R	R	R
(V.)MPG.BIDIR[m]	表格 [m]. 双向补偿 "0" = 否 "1"= 是	R	R	R
(V.)MPG.REFNEED[m]	表格 [m]. 强制机床零点搜寻 "0" = 否 "1"= 是	R	R	R
(V.)MPG.POSITION[m][i]	表格 [m]. 点 [i] 的主控轴位置	R	R	R
(V.)MPG.POSERROR[m][i]	表格 [m]. 正方向点 [i] 误差	R	R	R
(V.)MPG.NEGERROR[m][i]	表格 [m]. 负方向点 [i] 误差	R	R	R

执行时间		PRG	PLC	INT
(V.)MPG.MINAENDW	AUXEND 信号的最小持续时间	R	R	R
(V.)MPG.REFTIME	估计机床零点搜寻时间	R	R	R
(V.)MPG.HTIME	估计 "H" 功能时间	R	R	R
(V.)MPG.DTIME	估计 "D" 功能时间	R	R	R
(V.)MPG.TTIME	估计 "T" 功能时间	R	R	R

数字 I/O 的编号		PRG	PLC	INT
(V.)MPG.NDIMOD	数字输入模块的总数	R	R	R
(V.)MPG.NDOMOD	数字输出模块的总数	R	R	R
(V.)MPG.DIMODADDR[n]	数字输入模块的基础索引	R	R	R
(V.)MPG.DOMODADDR[n]	数字输出模块的基础索引	R	R	R

探针		PRG	PLC	INT
(V.)MPG.PROBE	有探针用于刀具校准。 "0" = 否 "1"= 是	R	R	R
(V.)MPG.PRBDI1	与探针 1 相关联的数字输入	R	R	R
(V.)MPG.PRBDI2	与探针 2 相关联的数字输入	R	R	R
(V.)MPG.PRBPULSE1	探针 1 的脉冲类型 "0" = 负 "1" = 正	R	R	R
(V.)MPG.PRBPULSE2	探针 2 的脉冲类型 "0" = 负 "1" = 正	R	R	R

15.2.1 与通道相关的变量

这些变量为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

• 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
• 使用轴号替代字母 “x”。

(V.)[n].MPG.GROUPID	V.[1].MPG.GROUPID	V.[2].MPG.GRUOPID
(V.)[n].MPG.CHAXISNAMEx	V.[2].MPG.CHAXISNAME2	V.[1].MPG.CHAXISNAME3



通道配置		PRG	PLC	INT
(V.)[n].MPG.GROUPID	通道所属的组	R	R	R
(V.)[n].MPG.CHTYPE	通道类型 "0" = CNC "1" = PLC "2" = CNC+PLC	R	R	R
(V.)[n].MPG.HIDDENCH	隐藏通道 "0" = 否 "1"= 是	R	R	R

配置通道轴		PRG	PLC	INT
(V.)[n].MPG.CHNAXIS	通道轴的数量	R	R	R
(V.)[n].MPG.CHAXISNAMEx	"n" 逻辑轴的命名	—	—	R
(V.)[n].MPG.GEOCONFIG	通道轴的几何学配置 "0" = 平面 "1" = 三面体	R	R	R

配置通道主轴		PRG	PLC	INT
(V.)[n].MPG.CHNSPDL	通道主轴的数量	R	R	R
(V.)[n].MPG.CHSPDLNAMEx	"x" 主轴的命名	—	—	R

C 轴的配置		PRG	PLC	INT
(V.)[n].MPG.CAXNAME	如 "C" 轴一样工作的轴 (缺省值)	—	—	R
(V.)[n].MPG.ALIGNC	径向加工的 "C" 轴 "0" = 否 "1"= 是	R	R	R

时间设置 (通道)		PRG	PLC	INT
(V.)[n].MPG.PREPFREQ	每个循环要准备的程序段数量	R	R	R
(V.)[n].MPG.ANTIME	预期时间	R	R	R

缺省条件 (通道)		PRG	PLC	INT
(V.)[n].MPG.KINID	缺省的运动数量	R	R	R
(V.)[n].MPG.SLOPETYPE	缺省的加速度类型 "1" = 线性 "2" = 梯形 "3" = 方形 - 正弦形	R	R	R
(V.)[n].MPG.IPLANE	缺省工作平面 "0" = G17 "1" = G18	R	R	R
(V.)[n].MPG.ISYSTEM	缺省编程类型 "0" = G90 "1" = G91	R	R	R
(V.)[n].MPG.IMOVE	缺省运动类型 "0" = G00 "1" = G01	R	R	R
(V.)[n].MPG.IFEED	缺省进给率类型 "0" = G94 "1" = G95	R	R	R
(V.)[n].MPG.FPRMAN	jog 模式是否允许使用功能 G95 "0" = 否 "1"= 是	R	R	R
(V.)[n].MPG.ICORNER	缺省拐角类型 "0" = G50 "1" = G05 "2" = G07	R	R	R
(V.)[n].MPG.IRCOMP	缺省的半径补偿模式 "0" = G136 "1" = G137	R	R	R
(V.)[n].MPG.ROUNDTYPE	G5 模式下的倒角类型 (缺省) "0" = 弦误差 "1" = % 进给率	R	R	R
(V.)[n].MPG.MAXROUND	G5 模式下最大倒角误差	R	R	R
(V.)[n].MPG.ROUNDFEED	G5 模式下的进给率百分率	R	R	R



圆心校正		PRG	PLC	INT
(V.)[n].MPG.CIRINERR	绝对半径误差	R	R	R
(V.)[n].MPG.CIRINFACT	超过半径误差的百分率	R	R	R

*CIRINFACT 的 PLC 读数*以十分之一（10 的读数为 %1）给出。参考 313 页 *“从 PLC 访问数字值”*。

进给率和进给率倍率的状态		PRG	PLC	INT
(V.)[n].MPG.MAXOVR	轴最大进给率 (%)	R	R	R
(V.)[n].MPG.RAPIDOVR	影响 G00 的倍率 "0" = 否 "1"= 是	R	R	R
(V.)[n].MPG.FEEDND	施加编程进给率给通道的所有轴 "0" = 否 "1"= 是	R	R	R

*MAXOVR 的 PLC 读数*以十分之一（10 的读数为 %1）给出。参考 313 页 *“从 PLC 访问数字值”*。

独立轴运动		PRG	PLC	INT
(V.)[n].MPG.IMOVEMACH	相对于机床坐标的独立轴运动	R	R	R

相关子程序		PRG	PLC	INT
(V.)[n].MPG.TOOLSUB	与 "T" 相关联的子程序	—	—	R
(V.)[n].MPG.REFPSUB	与 G74 相关联的子程序	—	—	R
(V.)[n].MPG.OEMSUB(1..10)	与 G180 到 G189 相关联的子程序	—	—	R
(V.)[n].MPG.SUBPATH	编辑子程序路径	—	—	R

探针		PRG	PLC	INT
(V.)[n].MPG.PRB1MIN	沿横坐标轴方向的探针最小坐标	R	R	R
(V.)[n].MPG.PRB1MAX	沿横坐标轴方向的探针最大坐标	R	R	R
(V.)[n].MPG.PRB2MIN	沿纵坐标轴方向的探针最小坐标	R	R	R
(V.)[n].MPG.PRB2MAX	沿纵坐标轴方向的探针最大坐标	R	R	R
(V.)[n].MPG.PRB3MIN	沿垂直于平面方向的轴的探针最小坐标	R	R	R
(V.)[n].MPG.PRB3MAX	沿垂直于平面方向的轴的探针最大坐标	R	R	R



15.3 与轴机床参数相关的变量

如果这些变量与主轴有关，可使用前缀 –MPA.– 或 –SP.– 访问它们。

可通过程序 (PRG)、PLC 和接口 (INT) 访问这些变量，它们为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 使用轴或主轴通道的名字、逻辑号或索引替代 “Xn” 字符。
- 保留括号，使用数字替代 “i” 字符。

(V.)[n].MPA.AXISTYPE.Xn	V.[1].MPA.AXISTYPE.X V.SP.AXISTYPE.S	V.[2].MPA.AXISTYPE.1 V.[3].SP.AXISTYPE.6
(V.)[n].MPA.INCJOGDIST[i].Xn	V.[2].MPA.INCJOGDIST[1].Z	V.[4].MPA.INCJOGDIST[2].3

通道属性		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.AXISEXCH	通道改变许可通道转换许可 "0" = 否 "1" = 暂时的 "2" = 持久的	是	是	是	是	P/S

轴和驱动类型		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DRIVETYPE.Xn	驱动器的类型驱动类型 "1" = 模拟 "2" = Sercos "16"= 仿真	是	是	是	是	P/S
(V.)[n].MPA.AXISTYPE.Xn	轴的类型轴类型 "1" = 线性轴 "2" = 旋转轴 "4" = 主轴	是	是	是	是	P/S
(V.)[n].MPA.DRIVEID.Xn	Sercos 驱动器选择 (ID)	是	是	是	—	P/S
(V.)[n].MPA.OPMODEP.Xn	Sercos 驱动操作模式 "0" = 位置 "1" = 速度	是	是	是	—	P/S
(V.)[n].MPA.FBACKSRC.Xn	轴的类型 "0" = 内部的 "1" = 外部的	是	是	是	—	P/S

HIRTH 轴		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.HIRTH.Xn	Hirth 轴 "0" = 否 "1"= 是	是	是	—	是	P/S
(V.)[n].MPA.HPITCH.Xn	Hirth 轴节距	是	是	—	是	P/S

车削类型机床轴的配置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.FACEAXIS.Xn	端面轴 "0" = 否 "1"= 是	是	—	—	是	P/S
(V.)[n].MPA.LONGAXIS.Xn	纵轴 "0" = 否 "1"= 是	是	—	—	是	P/S

轴和主轴的同步		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.SYNCSET.Xn	同步参数的设置	否	是	是	是	P/S
(V.)[n].MPA.DSYNCVELW.Xn	速度同步窗口	是	是	是	是	P/S
(V.)[n].MPA.DSYNCPOSW.Xn	位置同步窗口	是	是	是	是	P/S

旋转轴的配置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.AXISMODE.Xn	工作模式 "0" = 模块 "1" = 线性相似	—	是	—	是	P/S
(V.)[n].MPA.UNIDIR.Xn	单向旋转 "0" = 否 "1"= 是	—	是	—	是	P/S
(V.)[n].MPA.SHORTESTWAY.Xn	通过最短路线 "0" = 否 "1"= 是	—	是	—	是	P/S

旋转轴和主轴		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.MODCOMP.Xn	模块补偿 "0" = 否 "1"= 是	—	是	是	是	S
(V.)[n].MPA.CAXIS.Xn	如同 "C" 轴一样工作 "0" = 否 "1"= 是	—	是	是	是	P/S
(V.)[n].MPA.CAXSET.Xn	"C" 轴工作设置	—	是	是	是	P/S



主轴的配置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.AUTOGEAR.Xn	自动齿轮传动变换 "0" = 否 "1"= 是	—	—	是	是	P/S
(V.)[n].MPA.LOSPDLIM.Xn	"rpm OK" 百分率下限	—	—	是	是	P/S
(V.)[n].MPA.UPSPDLIM.Xn	"rpm OK" 百分率上限	—	—	是	是	P/S
(V.)[n].MPA.SPDLTIME.Xn	S 功能的估计时间	—	—	是	是	P/S
(V.)[n].MPA.SPDLSTOP.Xn	M2, M30 和复位停止主轴 "0" = 否 "1"= 是	—	—	是	是	P/S
(V.)[n].MPA.SREVM05.Xn	G84 反转停止主轴 "0" = 否 "1"= 是	—	—	是	是	P/S
(V.)[n].MPA.STEPOVR.Xn	倍率步幅	—	—	是	是	P/S
(V.)[n].MPA.MINOVR.Xn	最小倍率 (%)	—	—	是	是	P/S
(V.)[n].MPA.MAXOVR.Xn	最大倍率 (%)	—	—	是	是	P/S

LOSPDLIM, *UPSPDLIM*, *STEPOVR*, *MINOVR* 和 *MAXOVR 的 PLC 读数*以十分之一（10 的读数为 %1）给出。参考 313 页 *"从 PLC 访问数字值"*。

软件轴极限		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.POSLIMIT.Xn	正向软件极限	是	是	—	是	P/S
(V.)[n].MPA.NEGLIMIT.Xn	负向软件极限	是	是	—	是	P/S
(V.)[n].MPA.SWLIMITTOL.Xn	软件极限公差	是	是	—	是	P/S

失控保护		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.TENDENCY.Xn	趋向检测激活 "0" = 否 "1"= 是	是	是	是	是	P/S

PLC 偏置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.PLCOINC.Xn	每个循环 PLC 的偏置增量	是	是	是	是	P/S

死轴暂停		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DWELL.Xn	死轴停留时间	是	是	是	是	P/S

半径 / 直径		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DIAMPROG.Xn	以直径编程 "0" = 否 "1"= 是	是	—	—	是	P/S

机床原点搜寻		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.REFDIREC.Xn	原点搜寻方向 "0" = 负向 "1" = 正向	是	是	是	是	P/S
(V.)[n].MPA.DECINPUT.Xn	原点开关 "0" = 否 "1"= 是	是	是	是	是	P/S

探测运动的配置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.PROBEAXIS.Xn	探测轴	是	是	—	是	P/S
(V.)[n].MPA.PROBERANGE.Xn	最大制动距离	是	是	—	是	P/S
(V.)[n].MPA.PROBEFEED.Xn	探测进给率	是	是	—	是	P/S
(V.)[n].MPA.PROBEDELAY	" 探针 1" 信号的延迟	是	是	—	是	P/S
(V.)[n].MPA.PROBEDELAY	" 探针 2" 信号的延迟	是	是	—	是	P/S

刀具检验		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.REPOSFEED.Xn	最大复位进给率	是	是	—	是	P/S

独立轴的配置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.POSFEED.Xn	定位进给率	是	是	是	是	P/S



JOG 模式		Lin Rot	Spd	Ana	Ser
(V.)[n].MPA.MANPOSSW.Xn	G201 的最大正向行程	是	—	是	P/S
(V.)[n].MPA.MANNEGSW.Xn	G201 的最大负向行程	是	—	是	P/S
(V.)[n].MPA.JOGFEED.Xn	连续 JOG 模式进给率	是	—	是	P/S
(V.)[n].MPA.JOGRAPFEED.Xn	连续 JOG 模式下的快速进给	是	—	是	P/S
(V.)[n].MPA.MAXMANFEED.Xn	连续 JOG 模式下的最大进给	是	—	是	P/S
(V.)[n].MPA.MAXMANACC.Xn	JOG 模式下的最大加速度	是	—	是	P/S
(V.)[n].MPA.MANFEEDP.Xn	G201 的 jog 进给率的最大值 %	是	—	是	P/S
(V.)[n].MPA.IPOFEEDP.Xn	G201 的执行进给率的最大值 %	是	—	是	P/S
(V.)[n].MPA.MANACCP.Xn	G201 的 jog 加速度的最大值 %	是	—	是	P/S
(V.)[n].MPA.IPOACCP.Xn	G201 的执行加速度的最大值 %	是	—	是	P/S

JOG 模式 . 手轮		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.MPGRESOL[i].Xn	[i] 位置的刻度盘分辨率	是	是	—	是	P/S
(V.)[n].MPA.MPGFILTER.Xn	手轮的过滤时间	是	是	—	是	P/S

JOG 模式 . 增量 JOG		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.INCJOGDIST[i].Xn	[i] 刻度盘位置的运动距离	是	是	—	是	P/S
(V.)[n].MPA.INCJOGFEED[i].Xn	[i] 位置的进给率	是	是	—	是	P/S

丝杠误差补偿		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.LSCRWCOMP.Xn	丝杠误差补偿 "0" = 否 "1"= 是	是	是	是	是	P/S
(V.)[n].MPA.NPOINTS.Xn	表格中点的数量	是	是	是	是	P/S
(V.)[n].MPA.TYPLSCRW.Xn	补偿的类型 "0" = 实际坐标 "1" = 理论坐标	是	是	是	是	P/S
(V.)[n].MPA.BIDIR.Xn	双向补偿 "0" = 否 "1"= 是	是	是	是	是	P/S
(V.)[n].MPA.REFNEED.Xn	强制原点搜寻 "0" = 否 "1"= 是	是	是	是	是	P/S
(V.)[n].MPA.POSITION[i].Xn	点 [i] 的主控轴位置	是	是	是	是	P/S
(V.)[n].MPA.POSERROR[i].Xn	点 [i] 的正向误差	是	是	是	是	P/S
(V.)[n].MPA.NEGERROR[i].Xn	点 [i] 的负向误差	是	是	是	是	P/S

消除频率的滤波器		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.ORDER[i].Xn	滤波器指令	是	是	是	是	P/S
(V.)[n].MPA.TYPE[i].Xn	滤波器类型 "1" = 低通 "2" = 反共振	是	是	是	是	P/S
(V.)[n].MPA.FREQUENCY[i].Xn	转折或中心频率	是	是	是	是	P/S
(V.)[n].MPA.NORBWIDTH[i].Xn	标准带宽	是	是	是	是	P/S
(V.)[n].MPA.SHARE[i].Xn	通过滤波器信号的百分率	是	是	是	是	P/S

工作设置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.NPARSETS.Xn	工作设置数量	是	是	是	是	P/S
(V.)[n].MPA.DEFAULTSET.Xn	缺省工作设置 (通电状态)	是	是	是	是	P/S



15.3.1 与换档参数相关的变量

可以通过程序 (PRG)、PLC 和接口 (INT) 访问这些变量，它们为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 保留括号，使用齿轮编号替代 “g” 字母。使用编号 1 来识别第一个齿轮，“0” 为无效编号。
- 使用轴或主轴通道的名字、逻辑号或索引替代 “Xn” 字符。

(V.)[n].MPA.COUNTERID[g].Xn	V.[1].MPA.COUNTERID[1].X	V.[2].MPA.COUNTERID[2].1
(V.)[n].MPA.PITCH[g].Xn	V.[2].MPA.PITCH[1].Z	V.[4].MPA.PITCH[2].3

反馈分辨率		Lin Rot	Spd	Ana	Ser
(V.)[n].MPA.PITCH[g].Xn	丝杠螺距	是	是	是	P/S
(V.)[n].MPA.PITCH2[g].Xn	丝杠螺距 (二阶反馈)	是	是	—	P/S
(V.)[n].MPA.NPULSES[g].Xn	编码器脉冲数量	是	是	是	S
(V.)[n].MPA.NPULSES2[g].Xn	编码器脉冲数量 (二阶反馈)	是	是	是	S
(V.)[n].MPA.INPUTREV[g].Xn	电机轴转数	是	是	是	P/S
(V.)[n].MPA.INPUTREV2[g].Xn	电机轴转数 (二阶反馈)	是	是	—	P/S
(V.)[n].MPA.OUTPUTREV[g].Xn	机床轴转数	是	是	是	P/S
(V.)[n].MPA.OUTPUTREV2[g].Xn	机床轴转数 (二阶反馈)	是	是	—	P/S
(V.)[n].MPA.SINMAGNI[g].Xn	正弦曲线倍乘因子	是	是	是	—
(V.)[n].MPA.ABSFEEDBACK[g].Xn	绝对反馈系统 "0" = 否 "1"= 是	是	是	是	P/S
(V.)[n].MPA.FBACKAL[g]	反馈警报激活 "0" = 否 "1"= 是	是	是	是	—

循环设置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.LOOPCH[g].Xn	模拟电压信号转换 "0" = 否 "1"= 是	是	是	是	是	P/S
(V.)[n].MPA.AXISCH[g].Xn	反馈信号转换 "0" = 否 "1"= 是	是	是	是	是	P/S
(V.)[n].MPA.INPOSW[g].Xn	在适当位置区域	是	是	是	是	P/S

运动反转间隙		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.BACKLASH[g].Xn	背隙	是	是	是	是	P/S

背隙 . 附加速度指令脉冲		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.BAKANOUT[g].Xn	附加速度指令脉冲	是	是	是	是	P/S
(V.)[n].MPA.BAKTIME[g].Xn	附加速度指令脉冲持续时间	是	是	是	是	P/S
(V.)[n].MPA.ACTBAKAN[g].Xn	附加速度指令脉冲的应用 "0" = G2/G3 "1" = 总是	是	是	是	是	P/S

进给率设置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.G00FEED[g].Xn	G00 方式下的进给率	是	是	是	是	P/S
(V.)[n].MPA.MAXVOLT[g].Xn	G00FEED 的模拟电压	是	是	是	是	S



增益设置		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.PROGAIN[g].Xn	比例增益	是	是	是	是	P/S
(V.)[n].MPA.FFWTYPE[g].Xn	预先控制 （前馈）类型 "0" = 关闭 "1" = 前馈 "2" = Ac- 前馈 "3" = 前馈 + Ac- 前馈	是	是	是	是	P/S
(V.)[n].MPA.FFGAIN[g].Xn	自动模式下前馈的百分率	是	是	是	是	P/S
(V.)[n].MPA.MANFFGAIN[g].Xn	JOG 模式下前馈的百分率	是	是	—	是	P/S
(V.)[n].MPA.ACFWFACTOR[g].Xn	加速度时间常数	是	是	是	是	S
(V.)[n].MPA.ACFGAIN[g].Xn	自动模式下 AC- 前馈的百分率	是	是	是	是	P/S
(V.)[n].MPA.MANACFGAIN[g].Xn	JOG 模式下 AC- 前馈的百分率	是	是	—	是	P/S

虽然在机床参数表格中，它们的读数可带多达四位的小数，但是在任意情形下，以下变量的读数只能带一或两位小数。

- *变量 ACFGAIN 和 MANACFGAIN 中， 只有第一位小数有效。*
- *变量 FFGAIN 和 MANFFGAIN 中，前两位小数有效。*

*ACFGAIN*和*MANACFGAIN 的PLC*读数以十分之一(x10)给出。*FFGAIN*和*MANFFGAIN 的PLC*读数以百分之一 (x100) 给出。参考 313 页 *“从 PLC 访问数字值”*。

线性加速度		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.LACC1[g].Xn	第一部分加速度	是	是	是	是	P/S
(V.)[n].MPA.LACC2[g].Xn	第二部分加速度	是	是	是	是	P/S
(V.)[n].MPA.LFEED[g].Xn	转变速度	是	是	是	是	P/S

梯形和方波 – 正弦型加速度		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.ACCEL[g].Xn	加速度	是	是	是	是	P/S
(V.)[n].MPA.DECEL[g].Xn	减速度	是	是	是	是	P/S
(V.)[n].MPA.ACCJERK[g].Xn	加加速度	是	是	是	是	P/S
(V.)[n].MPA.DECJERK[g].Xn	减减速度	是	是	是	是	P/S

原点搜寻		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.I0TYPE[g].Xn	参考标志 (I0) 类型 "0" = 标准 "1" = 渐增距离编码 "2" = 渐减距离编码	是	是	是	是	P/S
(V.)[n].MPA.REFVALUE[g].Xn	原点位置	是	是	是	是	P/S
(V.)[n].MPA.REFSHIFT[g].Xn	参考点 （原点）偏置	是	是	是	是	P/S
(V.)[n].MPA.REFFEED1[g].Xn	快速原点搜寻进给率	是	是	是	是	P/S
(V.)[n].MPA.REFFEED2[g].Xn	慢速原点搜寻进给率	是	是	是	是	P/S
(V.)[n].MPA.REFPULSE[g].Xn	I0 脉冲的类型 "0" = 负 "1" = 正	是	是	是	是	P/S
(V.)[n].MPA.ABSOFF[g].Xn	关于编码参考标志的偏置	是	是	是	是	P/S
(V.)[n].MPA.EXTMULT[g].Xn	距离编码标志的外部因素	是	是	是	是	—
(V.)[n].MPA.I0CODDI1[g].Xn	2 个固定编码标志之间的间距	是	是	是	是	P/S
(V.)[n].MPA.I0CODDI2[g].Xn	2 个变量编码标志之间的间距	是	是	是	是	P/S

跟随误差		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.FLWEMONITOR[g].Xn	监控类型 "0" = 关闭 "1" = 标准 "2" = 线性	是	是	是	是	P/S
(V.)[n].MPA.MINFLWE[g].Xn	停止时的最大跟随误差	是	是	是	是	P/S
(V.)[n].MPA.MAXFLWE[g].Xn	运动时的最大跟随误差	是	是	是	是	P/S
(V.)[n].MPA.FEDYNFAC[g].Xn	跟随误差偏差的 %	是	是	是	是	P/S
(V.)[n].MPA.ESTDELAY[g].Xn	跟随误差延迟	是	是	是	是	P/S
(V.)[n].MPA.INPOMAX[g].Xn	到达恰当位置的时间	是	是	是	是	P/S
(V.)[n].MPA.INPOTIME[g].Xn	停留在恰当位置的最小时间	是	是	是	是	P/S

轴的润滑		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DISTLUBRI[g].Xn	关于润滑脉冲的时间间隔	是	是	是	是	P/S

旋转轴和主轴		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.MODUPLIM[g].Xn	模块的上限	—	是	是	是	S
(V.)[n].MPA.MODLOWLIM[g].Xn	模块的下限	—	是	是	是	S
(V.)[n].MPA.MODNROT[g].Xn	模块误差 . 转数	—	是	是	是	S
(V.)[n].MPA.MODERR[g].Xn	模块误差 . 增量数	—	是	是	是	S



主轴		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.SZERO[g].Xn	被认为是 "0 转 / 分 " 的速度	—	—	是	—	P/S
(V.)[n].MPA.POLARM3[g].Xn	模拟电压符号 M3 "0" = 负 "1" = 正	—	—	是	—	S
(V.)[n].MPA.POLARM4[g].Xn	模拟电压符号 M4 "0" = 负 "1" = 正	—	—	是	—	S

模拟电压		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.SERVOOFF[g].Xn	偏置补偿	是	是	是	是	—
(V.)[n].MPA.MINANOUT[g].Xn	最小模拟输出	是	是	是	是	—

模拟输出 / 反馈输入		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.ANAOUTID[g].Xn	轴的模拟输出	是	是	是	是	—
(V.)[n].MPA.COUNTERID[g].Xn	轴的反馈输入	是	是	是	是	—

15.4 与 jog 模式参数相关的变量

这些变量为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

• 保留括号，使用数字替代 “i” 字母。

(V.)MPMAN.NMPG	V.MPMAN.NMPG	
(V.)MPMAN.MPGAXIS[i]	V.MPMAN.MPGAXIS[1]	V.MPMAN.MPGAXIS[2]

手轮		PRG	PLC	INT
(V.)MPMAN.NMPG	手轮的数量	R	R	R
(V.)MPMAN.COUNTERID[i]	手轮 [i] 的反馈输入	R	R	R
(V.)MPMAN.MPGAXIS[i]	与手轮 [i] 关联的轴	R	R	R

JOG 键		PRG	PLC	INT
(V.)MPMAN.JOGKEYDEF[i]	JOG [i] 键的轴和运动方向	R	R	R
(V.)MPMAN.JOGTYPE	JOG 状态	R	R	R

该变量可取如下值:

"1", "2"..."16" = 机床参数设置成 "+1", "+2"..."+16". (轴和正方向使用的键)

"-1", "-2"... "-16" = 机床参数设置成 "-1", "-2"..."-16". (轴和负方向使用的键)

"101", "102"..."116" = 机床参数设置成 "1", "2"..."16". (轴键)

"300" = 机床参数设置成 "R". (快速运动键)

"301" = 机床参数设置成 "+". (正方向使用的键)

"302" = 机床参数设置成 "-". (负方向使用的键)

15.5 与 "M" 功能参数相关的变量

这些变量为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

- 保留括号，使用数字替代 “i” 字母。

(V.)MPM.MNUM[i]	V.MPM.MNUM[3]
(V.)MPM.MTABLESIZE	V.MPM.MTABLESIZE

"M" 功能表格		PRG	PLC	INT
(V.)MPM.MTABLESIZE	"M" 功能表格元素的数量	R	R	R
(V.)MPM.MNUM[i]	"M" 功能号	R	R	R
(V.)MPM.SYNCHTYPE[i]	"M" 功能同步类型 "0" = 不同步 "2" = 前-前 "4" = 前-后 "8" = 后-后	R	R	R
(V.)MPM.MTIME[i]	"M" 功能估计时间	R	R	R
(V.)MPM.MPROGNAME[i]	与 "M" 功能相关的子程序名	—	—	R





15.6 运动学相关参数

这些变量是只读 (R) 同步，在执行时被计算。

它们有类属名。

- 用运动学编码替代字母 "n" 。
- 用偏置编码替代字母 "m"

(V.)MPK.KINn[m]	V.MPK.KIN1[1]	V.MPK.KIN6[42]

运动学		PRG	PLC	INT
(V.)MPK.NKIN	运动学表	R	R	R
(V.)MPK.TYPE	运动学类型	R	R	R
(V.)MPK.KINn[m]	运动学 "n" 的偏置 [m]	R	R	R

角变换		PRG	PLC	INT
(V.)MPK.NANG	角变换编码	R	R	R
(V.)MPK.ANGAXNA[n]	角度轴名称	R	R	R
(V.)MPK.ORTAXNA[n]	正交轴名称	R	R	R
(V.)MPK.ANGANTR[n]	笛卡尔轴和倾斜轴之间的夹角	R	R	R
(V.)MPK.OFFANGAX[n]	角度变换起点偏置	R	R	R



15.7 与刀库参数相关的变量

这些变量为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

- 使用刀库编号替代字母 “z”，保留括号。

(V.)TM.MZSIZE[z]	V.TM.MZSIZE[1]

刀库		PRG	PLC	INT
(V.)TM.NTOOLMZ	刀库编号	R	R	R
(V.)TM.MZGROUND[z]	允许使用基础刀具 "0" = 否 "1"= 是	R	R	R
(V.)TM.MZSIZE[z]	刀库尺寸	R	R	R
(V.)TM.MZRANDOM[z]	随机刀库 "0" = 否 "1"= 是	R	R	R
(V.)TM.MZTYPE[z]	刀库类型 "1" = 异步 "2" = 同步 "3" = 转塔式 "4" = 与换刀臂 1 同步 "5" = 与换刀臂 2 同步	R	R	R
(V.)TM.MZCYCLIC[z]	循环换刀架 "0" = 否 "1"= 是	R	R	R
(V.)TM.MZOPTIMIZED[z]	刀具管理 "0" = 否 "1"= 是	R	R	R
(V.)TM.MZM6ALONE[z]	在无刀具时执行 M6 的动作 "0" = 无 "1" = 报警 "2" = 出错	R	R	R



15.8 与 OEM 参数相关的变量

这些变量为只读 (R) 同步，在执行过程中进行赋值。

它们有通用的名字。

- 保留括号，使用参数编号替代字母 “i” 。该编号与在机床参数表格中的参数编号一致。例如，出现在 MTBPAR 表格中的 P0 参数将作为 (V.)MTB.P[0] 被访问。

(V.)MTB.P[i]	V.MTB.P[3]

共享存储器		PRG	PLC	INT
(V.)MTB.PLCDATASIZE	PLC 的共享数据空间尺寸	R	R	R

OEM 参数		PRG	PLC	INT
(V.)MTB.SIZE	OEM 参数的数量	R	R	R
(V.)MTB.P[i]	OEM 参数 [i] 的值	R	R	R
(V.)MTB.PF[i]	OEM 参数 [i] 的值，每 10000 的值	R	R	R

当从 PLC 读取变量 (V.)MTB.P[i] 时，将截去小数部分。变量 (V.)MTB.PF[i] 返回乘 10000 的参数值。

```
DATA = 54.9876
(V.)MTB.P[10] = 54
(V.)MTB.PF[10] = 549876
```

读取驱动器变量		PRG	PLC	INT
(V.)DRV.SIZE	驱动器查询的变量数	R	R	R
(V.)DRV.name	变量值	R/W	R/W	R/W

驱动器变量的访问可为读或写，这取决于它在机床参数表中的设置。同样地，从 PLC 访问这些变量的类型 （同步或异步）也在机床参数表中定义。





15.9 与用户表格相关的变量

这些变量为读 / 写 (R/W) 同步，在执行过程中进行赋值。

它们有通用的名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 使用数字替代字母 "m" 和 "i"，保留括号。
- 用轴通道名称、逻辑号或索引替代字符 "Xn" 。

(V.)A.ORGT[i].Xn	V.A.ORGT[1].X	V.A.ORGT[1].1
(V.)A.FIX.Xn	V.A.FIX.X	V.A.FIX.2
(V.)G.LUPm[n]	V.G.LUP2[12]	

零点偏置表		Lin Rot	Spd	PRG	PLC	INT	Exec
(V.)G.FORG	表中第一零点偏置	—	—	R	R	R	是
(V.)G.NUMORG	表中零点偏置编码	—	—	R	R	R	是
(V.)[n].A.ORG.Xn	Xn 轴当前原点偏置	是	否	R	R	R	否
(V.)[n].A.ORGT[i].Xn	Xn 轴 [i] 原点偏置	是	否	R/W	R/W	R/W	是
(V.)[n].A.PLCOF.Xn	Xn 轴 PLC 原点偏置	是	否	R/W	R/W	R	是
(V.)[n].A.ACTPLCOF.Xn	在 Xn 轴累计 PLC 偏置	是	是	R	R	R	是

零点偏置的编码从 G54 到 G59 总是相同的：

G54=1, G55=2, G56=3, G57=4, G58=5, G59=6

ZERO'S OFFSETS

Origin	X (mm)	Y (mm)	Z (mm)	U (mm)	V (mm)
PLCOF	00000.000	00000.000	00000.000	00000.000	00000.000
G54	00000.000	00000.000	00000.000	00000.000	00000.000
G55	00000.000	00000.000	00000.000	00000.000	00000.000
G56	00000.	000		0000.000	00000.000
G57	00000.000	00000.000	00000.000	00000.000	00000.000
G58	00000.000	00000.000	00000.000	00000.000	00000.000
G59	00000.000	00000.000	00000.000	00000.000	00000.000
G159=7	00000.000	0000.000	00000.000	00000.000	00000.000
G159=8	00000.000		000.000	00000.000	00000.000
G159=9	00000.000	00000.000	00000.000	00000.000	00000.000
G159=10	00000.000	00000.000	00000.000	00000.000	00000.000
G159=11	00000.000	00000.000	00000.000	00000.000	00000.000
G159=12	00000.000	00000.000	00000.000	00000.000	00000.000
G159=13	00000.000	00000.000	00000.000	00000.000	00000.000

(V.)G.FROG

(V.)A.PLCOF.Y

(V.)A.ORGT[5].X



夹具表		Lin Rot	Spd	PRG	PLC	INT	Exec
(V.)G.FFIX	表中第一夹具	—	—	R	R	R	是
(V.)G.NUMFIX	表中夹具编码	—	—	R	R	R	是
(V.)[n].G.FIX	当前夹具编码	—	—	R/W	R	R	否
(V.)[n].A.FIX.Xn	轴当前夹具偏置	是	否	R	R	R	否
(V.)[n].A.FIXT[i].Xn	Xn 轴 [i] 夹具偏置	是	否	R/W	R/W	R/W	是

FIXTURE'S OFFSETS

Fixture	X (mm)	Y (mm)	Z (mm)	U (mm)	V (mm)
1	00000.000	00000.000	00000.000	00000.000	00000.000
2		00000.000	00000.000	00000.000	00000.000
3	00000.000	00000.000	00000.000	00000.000	00000.000
4	00000.000	00000.000	00000.000	00000.000	00000.000
5	00000.000	00000.000	00000.000		00000.000
6	00000.000	00000.000	00000.000		00000.000
7	00000.000	00000.000	00000.000	00000.000	00000.000
8			00000.000	00000.000	00000.000
9	00000.000	00000.000	00000.000	00000.000	00000.000
10	00000.000	00000.000	00000.000	00000.000	00000.000

(V.)G.FFIX
(V.)G.NUMFIX
(V.)A.FIXT[5].X

算数参数表		PRG	PLC	INT	Exec
(V.)G.CUP[i]	通用算数参数 [i] 值	—	R/W	R/W	是
(V.)G.CUPF[i]	通用算数参数 [i] 值 . 每 10000 值	—	R/W	R/W	是
(V.)[n].G.GUP[i]	全局算数参数 [i] 值	—	R/W	R/W	是
(V.)[n].G.GUPF[i]	全局算数参数 [i] 值 . 每 10000 值	—	R/W	R/W	是
(V.)[n].G.LUPACT[i]	局部算数参数 [i] 值 有效标准	—	R/W	R/W	是
(V.)[n].G.LUPm[i]	m 标准的局部算数参数 [i] 值	—	R/W	R/W	是
(V.)[n].G.LUPmF[i]	m 标准的局部算数参数 [i] 值 . 每 1000 值	—	R/W	R/W	是

当从 PLC 读取变量 G.CUP、G.GUP 以及 G.LUP1[i] ～ G.LUP7[i] 时，截去小数部分。变量 G.CUPF、G.GUPF 以及 G.LUP1F[i] ～ G.LUP7F[i] 返回乘 10000 后的参数值。

```
P100 = 23.1234
G.GUP[100] = 23
G.GUPF[100] = 231234
```




15.10 与刀具相关的变量

对于所有刀具变量，与有效刀具 (例如 TM.TOR) 相关的变量总是同步读取。与有效刀具无关的变量 (例如 TM.TORT[i][j])，如果刀具在刀库中，则支持同步读取，否则支持异步读取。这些变量的写入总是异步的，它也适合于有效刀具或非有效刀具。

刀具管理器变量的读取也是异步的。

这些变量在程序段执行期间被赋值，它们有通用名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 使用数字替代字母 "m" 、 "j" 和 "i"，保留括号。
- 用轴通道名称、逻辑号或索引替代字符 "Xn" 。

(V.)[n].TM.TOOL	V.[1].TM.TOOL	V.[4].TM.TOOL
(V.)TM.TORT[m][i]	V.TM.TORT[3][1]	V.TM.TORT[21][2]
(V.)TM.TOFLWT[m][i].Xn	(V.)TM.TOFLWT[4][1].X	(V.)TM.TOFLWT[4][1].1

刀具和刀具偏置		PRG	PLC	INT
(V.)TM.T[z][j]	[z] 刀库的在 [j] 位置 的刀具	R	R	R
(V.)TM.P[z][m]	在 [z] 刀库 [m] 刀具的位置	R	R	R
(V.)[n].TM.TOOL	有效刀具编码	R	R	R
(V.)[n].TM.TOD	有效刀具偏置编码	R	R	R
(V.)[n].TM.NXTOOL	下一刀具编码	R	R	R
(V.)[n].TM.NXTOD	下一刀具偏置编码	R	R	R

如果在变量 (V.)TM.T[z][j] 和 (V.)TM.P[z][m] 中， *[z]* 刀库的编号被遗漏，变量指向前一个。

" 下一把刀具 " 指已经选好，但是等待执行 M06 来激活的刀具。

监测		PRG	PLC	INT
(V.)[n].TM.TOMON[i]	有效刀具偏置 [i] 的监测类型	R	R	R
(V.)TM.TOMONT[m][i]	刀具 [m] 偏置 [i] 的监测类型	R/W	R/W	R/W
(V.)[n].TM.TLFN[i]	有效刀具偏置 [i] 的最大使用期限	R	R	R
(V.)TM.TLFNT[m][i]	刀具 [m] 偏置 [i] 的最大有效期限	R/W	R/W	R/W
(V.)[n].TM.TLFR[i]	有效刀具偏置 [i] 的实际寿命	R	R	R
(V.)TM.TLFRT[m][i]	刀具 [m] 偏置 [i] 的实际寿命	R/W	R/W	R/W
(V.)[n].TM.REMLIFE	有效刀具剩余使用寿命	—	R	R

如果在刀具变量中遗漏了偏置号，变量指向有效偏置。

刀库		PRG	PLC	INT
(V.)[n].TM.TSTATUS	有效刀具状态	R	R	R
(V.)TM.TSTATUST[m]	刀具 [m] 状态	—	R	R
(V.)[n].TM.TLFF	有效刀具谱	R	R	R
(V.)TM.TLFFT[m]	刀具 [m] 谱	R/W	R/W	R/W
(V.)[n].TM.ACTUALMZ	每个通道使用的刀库	R	R	R
(V.)TM.MZRESPECTSIZE[z]	在随机刀库 [z] 刀具总是在同一位置	R	R	R
(V.)TM.MZACTUALCH[z]	刀库 [z] 使用的通道	R	R	R



下列变量可以通过程序 (PRG)、 PLC 和接口 (INT) 访问，是读－写型 (R/W) 。

几何学 (铣刀)		Rot Lin	Spd
(V.)[n].TM.TOR[i]	有效刀具偏置 [i] 半径	—	—
(V.)TM.TORT[m][i]	刀具 [m] 偏置 [i] 半径	—	—
(V.)[n].TM.TOI[i]	有效刀具补偿 [i] 半径磨损	—	—
(V.)TM.TOIT[m][i]	刀具 [m] 补偿 [i] 半径磨损	—	—
(V.)[n].TM.TOL[i]	有效刀具长度偏置 [i]	—	—
(V.)TM.TOLT[m][i]	刀具 [m] 偏置 [i] 长度	—	—
(V.)[n].TM.TOK[i]	有效刀具补偿 [i] 长度磨损	—	—
(V.)TM.TOKT[m][i]	刀具 [m] 补偿 [i] 长度磨损	—	—
(V.)[n].TM.TOTIPR[i]	有效刀具偏置 [i] 刀尖半径	—	—
(V.)TM.TOTIPRT[m][i]	刀具 [m] 偏置 [i] 刀尖半径	—	—
(V.)[n].TM.TOWTIPR[i]	有效刀具补偿 [i] 刀尖半径磨损	—	—
(V.)TM.TOWTIPRT[m][i]	刀具 [m] 补偿 [i] 刀尖半径磨损	—	—
(V.)[n].TM.TOCUTL[i]	有效刀具偏置 [i] 切削长度	—	—
(V.)TM.TOCUTLT[m][i]	刀具 [m] 偏置 [i] 切削长度	—	—
(V.)[n].TM.TOAN[i]	有效刀具偏置 [i] 穿透角度	—	—
(V.)TM.TOANT[m][i]	刀具 [m] 偏置 [i] 穿透角度	—	—
(V.)[n].TM.TOFL[i].Xn	有效刀具偏置 [i] 的 Xn 轴偏差	是	否
(V.)[n].TM.TOFL1	通道第一轴刀具的偏置	是	否
(V.)[n].TM.TOFL2	通道第二轴刀具的偏置	是	否
(V.)[n].TM.TOFL3	通道第三轴刀具的偏置	是	否
(V.)TM.TOFLT[m][i].Xn	刀具 [m] 偏置 [i] 的 Xn 轴偏差	是	否
(V.)[n].TM.TOFLW[i].Xn	有效刀具偏置 [i] 的 Xn 轴偏差	是	否
(V.)[n].TM.TOFLW1	通道第一轴刀具的磨损补偿	是	否
(V.)[n].TM.TOFLW2	通道第二轴刀具的磨损补偿	是	否
(V.)[n].TM.TOFLW3	通道第三轴刀具的磨损补偿	是	否
(V.)TM.TOFLWT[m][i].Xn	刀具 [m] 偏置 [i] 的 Xn 轴偏差磨损	是	否

如果在刀具变量中遗漏偏置号，变量指向有效偏置。

(V.)TM.TOR[i]	有效刀具半径 , 偏置 [i]。
(V.)TM.TOR	有效刀具半径 , 有效偏置。
(V.)TM.TORT[m][i]	刀具 [m] 偏置 [i] 的半径。
(V.)TM.TORT[m]	刀具半径 [m], 通道中有效偏置。

"CUSTOM" 数据		PRG	PLC	INT
(V.)[n].TM.TOTP1	有效刀具的附加参数 1	R/W	R/W	R/W
(V.)[n].TM.TOTP2	有效刀具的附加参数 2	R/W	R/W	R/W
(V.)[n].TM.TOTP3	有效刀具的附加参数 3	R/W	R/W	R/W
(V.)[n].TM.TOTP4	有效刀具的附加参数 4	R/W	R/W	R/W
(V.)TM.TOTP1T[i]	刀具 [i] 附加参数 1	R/W	R/W	R/W
(V.)TM.TOTP2T[i]	刀具 [i] 附加参数 2	R/W	R/W	R/W
(V.)TM.TOTP3T[i]	刀具 [i] 附加参数 3	R/W	R/W	R/W
(V.)TM.TOTP4T[i]	刀具 [i] 附加参数 4	R/W	R/W	R/W

刀具管理器		PRG	PLC	INT
(V.)[n].TM.MZSTATUS	刀具管理器状态	—	R	R
(V.)[n].TM.MZRUN	刀具管理器运行	—	R	R
(V.)[n].TM.MZMODE	刀具管理器操作模式	R/W	R	R/W
(V.)[n].TM.MZWAIT	执行操纵的刀具管理器	R	R	R

(V.)TM.MZWAIT 在与 M06 关联的子程序中不需要编写它。子程序本身等待管理器的操纵完成。这样，程序段准备不会中断。



几何学 (车刀)		Rot Lin	Spd
(V.)[n].TM.NUMOFD	有效刀具偏置号	—	—
(V.)[n].TM.NUMOFDT[m]	刀具 [m] 偏置号	—	—
(V.)[n].TM.DTYPE[i]	刀具偏置类型 . 有效刀具偏置 [i]	—	—
(V.)[n].TM.DTYPET[i][m]	刀具偏置类型 . 刀具 [m] 偏置 [i]	—	—
(V.)[n].TM.DSUBTYPE[i]	刀具偏置子类型 . 有效刀具偏置 [i]	—	—
(V.)[n].TM.DSUBTYPET[i][m]	刀具偏置子类型 . 刀具 [m] 偏置 [i]	—	—
(V.)[n].TURNCONFIG[i]	有效刀具偏置 [i] 的轴设置	—	—
(V.)[n].TURNCONFIG[i][m]	刀具 [m] 偏置 [i] 的轴设置	—	—
(V.)[n].TM.LOCODE[i]	有效刀具偏置 [i] 位置码 (形状)	—	—
(V.)[n].TM.LOCODET[i][m]	刀具 [m] 偏置 [i] 位置码 (形状)	—	—
(V.)[n].TM.FIXORI[i]	有效刀具偏置 [i] 刀柄	—	—
(V.)[n].TM.FIXORIT[i][m]	刀具 [m] 偏置 [i] 刀柄	—	—
(V.)[n].TM.SPDLTURDIR[i]	主轴旋转方向 . 有效刀具偏置 [i]	—	—
(V.)[n].TM.SPDLTURDIRT[i][m]	主轴旋转方向 . 刀具 [m] 偏置 [i]	—	—
(V.)[n].TM.NOSEA[i]	有效刀具偏置 [i] 的刀具角	—	—
(V.)[n].TM.NOSEAT[i][m]	刀具 [m] 偏置 [i] 的刀具角	—	—
(V.)[n].TM.NOSEW[i]	有效刀具偏置 [i] 的刀具宽度	—	—
(V.)[n].TM.NOSEWT[i][m]	刀具 [m] 偏置 [i] 的刀具宽度	—	—
(V.)[n].TM.CUTA[i]	有效刀具偏置 [i] 的切削角	—	—
(V.)[n].TM.CUTAT[i][m]	刀具 [m] 偏置 [i] 的切削角	—	—
(V.)[n].TM.TOCUTL[i]	有效刀具偏置 [i] 的切削长度	—	—
(V.)TM.TOCUTLT[m][i]	刀具 [m] 偏置 [i] 的切削长度	—	—
(V.)[n].TM.TOTIPR[i]	有效刀具偏置 [i] 的刀尖半径	—	—
(V.)TM.TOTIPRT[m][i]	刀具 [m] 偏置 [i] 的刀尖半径	—	—
(V.)[n].TM.TOWTIPR[i]	有效刀具偏置 [i] 的刀尖半径磨损	—	—
(V.)TM.TOWTIPRT[m][i]	刀具 [m] 偏置 [i] 的刀尖半径磨损	—	—
(V.)[n].TM.TOFL[i].Xn	有效刀具偏置 [i] 的 Xn 轴距离	是	否
(V.)[n].TM.TOFL1	通道第一轴刀具的偏置	是	否
(V.)[n].TM.TOFL2	通道第二轴刀具的偏置	是	否
(V.)[n].TM.TOFL3	通道第三轴刀具的偏置	是	否
(V.)TM.TOFLT[m][i].Xn	刀具 [m] 偏置 [i] 的 Xn 轴距离	是	否
(V.)[n].TM.TOFLW[i].Xn	有效刀具偏置 [i] 的 Xn 轴磨损距离	是	否
(V.)[n].TM.TOFLW1	通道第一轴的刀具磨损补偿	是	否
(V.)[n].TM.TOFLW2	通道第二轴的刀具磨损补偿	是	否
(V.)[n].TM.TOFLW3	通道第三轴的刀具磨损补偿	是	否
(V.)TM.TOFLWT[m][i].Xn	刀具 [m] 偏置 [i] 的 Xn 轴磨损距离	是	否

(V.)TM.NUMOFD 为只读变量。



15.10.1 仅用在程序段准备期间的变量

为了预先计算随后的路径， CNC 预先读取正在执行的程序段之后的多个程序段。

从下例可以看出，被计算的准备程序段是与刀具 T6 相关，而当前选择的刀具为 T1。

```
G1 X100 F200 T1 M6        (执行中的程序段)
Y200
G1 X20 F300 T6 M6
X30 Y60                   (准备程序段)
```

在准备过程中，可使用查询和 / 或修改值的特殊变量。

只能通过程序 (PRG) 访问它们，在程序段准备期间进行赋值。

当写入这些变量中的任一个时，不修改表格；仅仅在程序段准备时才使用新值。

下表是关于准备刀具的，除非有另外说明。

仅在程序段准备期间使用		Lin	Rot	Spd	PRG
(V.)[n].G.TOOL	准备刀具号	—	—	—	R
(V.)[n].G.TOD	准备刀具偏置号	—	—	—	R
(V.)[n].G.NXTOOL	准备下一把刀具号	—	—	—	R
(V.)[n].G.NXTOD	准备下一把刀具偏置号	—	—	—	R
(V.)[n].G.TOR	准备刀具补偿半径	—	—	—	R/W
(V.)[n].G.TOI	准备刀具偏置的半径磨损	—	—	—	R/W
(V.)[n].G.TOL	准备刀具偏置的长度	—	—	—	R/W
(V.)[n].G.TOK	准备刀具偏置的长度磨损	—	—	—	R/W
(V.)[n].G.TOTIPR	准备偏置的刀尖半径	—	—	—	R/W
(V.)[n].G.TOWTIPR	准备刀具偏置的刀尖半径磨损	—	—	—	R/W
(V.)[n].G.TOCUTL	准备刀具偏置的切削长度	—	—	—	R/W
(V.)[n].G.TOAN	准备刀具偏置的穿透角度	—	—	—	R/W
(V.)[n].A.TOFL.Xn	Xn 轴有效偏置的偏差	是	是	否	R/W
(V.)[n].A.TOFLW.Xn	Xn 轴有效磨损补偿的偏差	是	是	否	R/W
(V.)[n].G.TOFL1	通道第一轴刀具的偏置	是	是	否	R/W
(V.)[n].G.TOFL2	通道第二轴刀具的偏置	是	是	否	R/W
(V.)[n].G.TOFL3	通道第三轴刀具的偏置	是	是	否	R/W
(V.)[n].G.TOFLW1	通道第一轴刀具的磨损补偿	是	是	否	R/W
(V.)[n].G.TOFLW2	通道第二轴刀具的磨损补偿	是	是	否	R/W
(V.)[n].G.TOFLW3	通道第三轴刀具的磨损补偿	是	是	否	R/W
(V.)[n].G.TOMON	准备刀具偏置的检测类型	—	—	—	R/W
(V.)[n].G.TLFN	准备刀具偏置的额定寿命	—	—	—	R
(V.)[n].G.TLFR	准备刀具偏置的实际寿命	—	—	—	R
(V.)[n].G.REMLIFE	准备刀具偏置的剩余寿命	—	—	—	R/W
(V.)[n].G.TSTATUS	准备刀具状态	—	—	—	R
(V.)[n].G.TLFF	准备刀具偏置的族	—	—	—	R
(V.)[n].G.TOTP1	激活刀具附加参数 1	—	—	—	R/W
(V.)[n].G.TOTP2	激活刀具附加参数 2	—	—	—	R/W
(V.)[n].G.TOTP3	激活刀具附加参数 3	—	—	—	R/W
(V.)[n].G.TOTP4	激活刀具附加参数 4	—	—	—	R/W



15.11 与 PLC 相关的变量

这些变量在程序段执行期间被赋值。

它们有通用名字。

- 使用数字替代 “i” 字符，保留括号。

(V.)PLC.I[n]	V.PLC.I[16]
(V.)PLC. 信号	V.PLC.auxend

状态		PRG	PLC	INT	R	W
(V.)PLC.STATUS	PLC 状态 "0" = 停止 "1" = 运行	R	—	R	—	—

资源		PRG	PLC	INT	R	W
(V.)PLC.I[i]	PLC 输入 [i] 状态	R/W	—	R/W	—	—
(V.)PLC.O[i]	PLC 输出 [i] 状态	R/W	—	R/W	—	—
(V.)PLC.M[i]	PLC 标志 [i] 状态	R/W	—	R/W	—	—
(V.)PLC.R[i]	PLC 寄存器 [i] 状态	R/W	—	R/W	—	—
(V.)PLC.T[i]	PLC 定时器 [i] 状态	R	—	R/W	—	—
(V.)PLC.C[i]	PLC 计数器 [i] 状态	R	—	R/W	—	—
(V.)PLC.signal	与 CNC 交换信号的状态 (任何的标志或寄存器)	R/W	—	R/W	—	—

符号		PRG	PLC	INT	R	W
(V.)PLC.symbol	在 PLC 中定义的外部符号的状态	R/W	—	R/W	—	—

该变量仅仅用于在 PLC 程序中查询用指令 PDEF 定义的符号。

信息		PRG	PLC	INT	R	W
(V.)PLC.MSG[i]	PLC 信息 [n] 状态 "0" = 无效 "1" = 有效	R/W	—	R/W	—	—
(V.)PLC.PRIORMSG	具有最高优先级的有效信息 (在有效信息中有最低编号的那个)	R	—	R	—	—
(V.)PLC.EMERGMSG	有效显示信息 (全屏显示)	R	—	R	—	—

错误		PRG	PLC	INT	R	W
(V.)PLC.ERR[i]	PLC 错误 [n] 状态 "0" = 无效 "1" = 有效	R/W	—	R/W	—	—
(V.)PLC.PRIORERR	具有最高优先级的活动错误 (在有效信息中有最低编号的那个)	R	—	R	—	—

定时器		PRG	PLC	INT	R	W
(V.)PLC.TIMER	PLC 激活的定时器值	R/W	R/W	R/W	Syn	Syn

用 PLC 标志 TIMERON 激活或取消 PLC " 定时器 " 。当 TIMERON=1 时计时，用变量 (V.)PLC.TIMER，可以查询和 / 或修改它的计时数，单位为秒。



15.12 与 Jog 模式相关的变量

使用操作面板上的 jog 选择开关，可以选择 " 运动类型 "， " 手轮分辨率 " 和 “增量 jog 定位”。

也可以通过 PLC 为其赋值。当从 PLC 设置值时， CNC 忽略选择器开关键。

为 X 轴手轮设置 "10" 位置实例：

设置变量 (V.)A.PLCMMODE.X 为 "1"
设置变量 (V.)PLC.MPGDIX 为 "2"

如果要 X 轴手轮 " 服从 " (不忽略) 开关：

设置变量 (V.)A.PLCMMODE.X 为 "0"。

这些变量可同步读 / 写 (R/W)。所有这些变量在执行时被赋值。

运动类型		Lin Rot	Spd	PRG	PLC	INT
(V.)G.MANMODE	对所有轴有效	—	—	R	R	R
(V.)G.CNCMANMODE	所有轴在开关处	—	—	R	R	R/W
(V.)PLC.MANMODE	所有轴通过 PLC	—	—	R	R/W	R
(V.)[n].A.MANMODE.Xn	对 Xn 轴有效	是	否	R	R	R
(V.)[n].A.CNCMMODE.Xn	Xn 轴在开关处	是	否	R	R	R/W
(V.)[n].A.PLCMMODE.Xn	Xn 轴通过 PLC	是	否	R	R/W	R

这些变量可以有下列值：

"0" = 没有从 PLC 获取类型。

"1" = 手轮模式。

"2" = 连续 jog 模式。

"3" = 增量 jog 模式。 .

变量 "(V.)[n].A.MANMODE.Xn" 也可以有下列值：

"4" = 未选择轴的手轮模式。手轮模式被选择，但是没有选择被运动的轴。

手轮模式分辨率 (位置)		PRG	PLC	INT
(V.)G.MPGIDX	对于所有手轮的有效位置	R	R	R
(V.)G.CNCMPGIDX	通过开关选择的位置	R	R	R/W
(V.)PLC.MPGIDX	通过 PLC 选择的位置	R	R/W	R

这些变量可以有下列值：

"1" = 位置 1

"2" = 位置 10

"3" = 位置 100

增量 JOG 位置		PRG	PLC	INT
(V.)G.INCJOGIDX	对所有轴的有效位置	R	R	R
(V.)G.CNCINCJOGIDX	通过开关选择的位置	R	R	R/W
(V.)PLC.INCJOGIDX	通过 PLC 选择的位置	R	R/W	R

这些变量可取下述值：

"1" = 位置 1

"2" = 位置 10

"3" = 位置 100

"4" = 位置 1000

"5" = 位置 10000



JOG 进给率		PRG	PLC	INT
(V.)[n].G.FMAN	G94 模式下的 JOG 进给率	R/W	R	R/W
(V.)[n].G.MANFPR	G95 模式下的 JOG 进给率	R/W	R	R/W

从 jog 模式界面改变 –F– 域的值时，与 jog 模式相关的变量被修改。从 MDI 模式改变进给率时这些变量不受影响。



15.13 与坐标相关的变量

记住，工作在闭环中的主轴 (M19 or G63) 与轴的行为类似。

这些变量可同步读取 (R)。

它们有通用名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 用轴通道名称、逻辑号或索引替代字符 "Xn" 。
- 用主轴通道中名称、逻辑号或索引替代字符 "Sn" 。

(V.)[n].A.PPOS.Xn	V.[1].A.PPOS.X	V.[1].A.PPOS.1
(V.)[n].A.POS.Sn	V.[2].A.POS.S	V.[2].A.POS.2

实际坐标和理论坐标对应刀具基准和刀尖，它们都可能是相对于机床零点或当前工件零点的坐标。

理论坐标是轴必须始终占据的位置，实际坐标是轴实际占据的位置，两者之差叫做 " 跟随误差 "。

与线性轴和旋转轴相关的变量		PRG	PLC	INT	Exec
(V.)[n].A.PPOS.Xn	编程坐标 (刀尖)	R	R	R	否
(V.)[n].G.PLPPOS1	编程坐标 (刀尖) 通道第一轴	R	R	R	否
(V.)[n].G.PLPPOS2	编程坐标 (刀尖) 通道第二轴	R	R	R	否
(V.)[n].G.PLPPOS3	编程坐标 (刀尖) 通道第三轴	R	R	R	否
(V.)[n].A.FLWE.Xn	轴的跟随误差	R	R	R	是
(V.)[n].A.FLWEST.Xn	轴跟随误差 (延迟) 的线性估算	R	R	R	是
(V.)[n].A.APOS.Xn	工件坐标，刀具基准实际坐标	R	R	R	是
(V.)[n].A.ATPOS.Xn	工件坐标，刀具基准理论坐标	R	R	R	是
(V.)[n].A.ATIPPOS.Xn	工件坐标，刀尖实际坐标	R	R	R	是
(V.)[n].A.ATIPTPOS.Xn	工件坐标，刀尖理论坐标	R	R	R	是
(V.)[n].A.POS.Xn	机床坐标，刀具基准实际坐标	R	R	R	是
(V.)[n].A.TPOS.Xn	机床坐标，刀具基准理论坐标	R	R	R	是
(V.)[n].A.TIPPOS.Xn	机床坐标，刀尖实际坐标	R	R	R	是
(V.)[n].A.TIPTPOS.Xn	机床坐标，刀尖理论坐标	R	R	R	是

PPOSS 变量返回在工件坐标、相对于刀尖与当前参考系统中的目标坐标；也就是说，考虑了坐标旋转、比例缩放因子、有效斜面等因素。

`G1 X10`	V.A.PPOS.X=10
`#SCALE [2]`	(比例缩放因子 ·2·)
`G1 X10`	V.A.PPOS.X=20
`G73 Q90`	[坐标系旋转 (图形旋转)]
`X10`	V.A.PPOS.Y=20 (因为 Y 轴是运动轴)

当坐标受刀具补偿影响或在圆角模式下加工时，从程序或 PLC 及接口读取的 PPOS 变量值将是不同的。通过程序读取的值将是编程坐标 ，但是从 PLC 或接口读取的值将是考虑刀具半径补偿和过渡圆角的实际坐标。

与主轴相关的变量		PRG	PLC	INT	Exec
(V.)[n].A.POS.Sn	实际主轴位置	R	R	R	否
(V.)[n].A.TPOS.Sn	理论主轴位置	R	R	R	是
(V.)[n].A.PPOS.Sn	编程主轴位置	R	R	R	是
(V.)[n].A.FLWE.Sn	主轴跟随误差	R	R	R	是
(V.)[n].A.FLWEST.Sn	主轴跟随误差 (延迟) 的线性估算	R	R	R	是



15.14 与进给率相关的变量

这些变量可同步读 / 写 (R/W)。

它们有通用名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。

进给率		PRG	PLC	INT	Exec
(V.)[n].G.FREAL	实际 CNC 进给率	R	R	R	是
(V.)[n].G.FEED	G94 模式下的有效进给率	R	R	R	是
(V.)[n].PLC.F	G94 模式下的 PLC 进给率	R	R/W	R	是
(V.)[n].G.PRGF	G94 模式下的编程进给率	R	R	R	否
(V.)[n].G.FPREV	G95 模式下的有效进给率	R	R	R	是
(V.)[n].PLC.FPR	G95 模式下的 PLC 进给率	R	R/W	R	是
(V.)[n].G.PRGFPR	G95 模式下的编程进给率	R	R	R	否

变量 (V.)G.FREAL 考虑机床的加速度和减速度。当轴停止时，返回 "0" 值；当轴运动时，返回对应于进给率类型 G94/G95 的值。在激光切割机上，推荐使用该变量，这样激光强度和进给率是成比例的。

在 G94 (mm/min) 或 G95 (mm/rev) 模式下的进给率可以通过编程或 PLC 来设置；由 PLC 设置的进给率具有最高优先级。在 MDI 模式下选择新的进给率时， CNC 更新下列变量：

- G94 有效时的 (V.)G.FEED 和 (V.)G.PRGF 变量。
- G95 有效时的 (V.)G.FPREV 和 (V.)G.PRGFPR 变量 。

加工时间		PRG	PLC	INT	Exec
(V.)G.FTIME	G93 模式下的加工时间	R	R	R	否
加工时间以秒为单位。					

进给率倍率		PRG	PLC	INT	Exec
(V.)[n].G.FRO	CNC 设置的有效进给率倍率 （ %）	R	R	R	是
(V.)[n].A.FRO.Xn	通过轴设置的有效进给率倍率 （ %）	R/W	R/W	R/W	是
(V.)[n].G.PRGFRO	通过程序设置的进给率倍率 （ %）	R/W	R	R	否
(V.)[n].PLC.FRO	通过 PLC 设置的进给率倍率 （ %）	R	R/W	R	是
(V.)[n].G.CNCFRO	使用选择开关设置的进给率倍率 （ %）	R	R	R/W	是

(V.)[n].A.FRO.Xn 对旋转轴和线性轴有效。

可以通过程序、 PLC 或选择开关设置进给率倍率 % ；编程设置的进给率倍率 % 有最高优先权，使用选择开关设置的进给率倍率 % 的优先权最低。



15.15 与主轴转速相关的变量

这些变量可同步读 / 写 (R/W)。

它们有通用名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 用主轴通道中的名称、逻辑号或索引替代字符 "Sn" 。

V.A.SREAL.Sn	V.A.SREAL.S

转速		PRG	PLC	INT	Exec
(V.)[n].A.SREAL.Sn	实际主轴转速	R	R	R	是

考虑主轴转速倍率。

Con el cabezal parado devuelve el valor 0. Trabajando en G96 y G97 la velocidad está en rpm y trabajando con M19 en º/min.

G97 模式下的主轴转速		PRG	PLC	INT	Exec
(V.)[n].A.SPEED.Sn	以 rpm 为单位的 (G97) 有效主轴转速	R	R	R	是
(V.)[n].PLC.S.Sn	PLC 设置的以 rpm 为单位的主轴转速	R	R/W	R	是
(V.)[n].A.PRGS.Sn	编程设置的以 rpm 为单位的主轴转速	R	R	R	否

速度可以通过编程或 PLC 来设置; 通过 PLC 设置的具有最高优先级。

CSS 模式下的主轴转速		PRG	PLC	INT	Exec
(V.)[n].A.CSS.Sn	有效 CSS	R	R	R	是
(V.)[n].PLC.CSS.Sn	通过 PLC 设置的 CSS	R	R/W	R	是
(V.)[n].A.PRGCSS.Sn	通过编程设置的 CSS	R	R	R	否

速度可以通过编程或 PLC 来设置; 通过 PLC 设置的具有最高优先级。

最大恒定表面速度		PRG	PLC	INT	Exec
(V.)[n].A.SLIMIT.Sn	在恒定表面速度模式下的有效表面速度极限	R	R	R	是
(V.)[n].PLC.SL.Sn	PLC 设置的在恒定表面速度模式下的表面速度极限	R	R/W	R	是
(V.)[n].A.PRGSL.Sn	编程设置的在恒定表面速度模式下的表面速度极限	R	R	R	否

当恒定表面速度有效时，这些变量仅限制主轴转速 (转数 / 分) 。最大恒定表面速度可以通过编程或 PLC 来设置； PLC 设置的最大恒定表面速度具有最高优先级。

主轴转速倍率		PRG	PLC	INT	Exec
(V.)[n].A.SSO.Sn	CNC 设置的有效主轴倍率 （%）	R	R	R	是
(V.)[n].A.PRGSSO.Sn	程序设置的主轴倍率 （%）	R/W	R	R	否
(V.)[n].PLC.SSO.Sn	PLC 设置的主轴倍率 （%）	R	R/W	R	是
(V.)[n].A.CNCSSO.Sn	通过选择开关选择的主轴倍率 （%）	R	R	R/W	是

主轴倍率设置可以通过编程、PLC 或选择按键完成；通过编程设置的具有最高优先级，通过选择按键设置的具有最低优先级。

M19 模式下的速度		PRG	PLC	INT	Exec
(V.)[n].A.SPOS.Sn	M19 方式下的有效转速	R	R	R	是
(V.)[n].PLC.SPOS.Sn	PLC 设置的在 M19 模式下的转速	R	R/W	R	是
(V.)[n].A.PRGSPOS.Sn	编程设置的在 M19 模式下的转速	R	R	R	否



15.16 与被编辑功能相关的变量

它们有通用名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 用轴通道名称、逻辑号或索引替代字符 "Xn" 。
- 使用数字替代 “i” 和 "x" 字符，保留括号。

这些变量可同步读取 (R)。

"G" 和 "M" 功能		PRG	PLC	INT	Exec
(V.)[n].G.GS[i]	被请求的 "G" 功能状态	R	R	R	否
(V.)[n].G.MS[i]	被请求的 "M" 功能状态	R	R	R	否
(V.)[n].G.HGS1..10	被请求的 "G" (32 位) 功能状态	R	R	R	否
(V.)[n].G.HGS	显示的 "G" 功能记录	—	—	R	是
(V.)[n].G.HMS	显示的主要主轴 "M" 功能记录	—	—	R	是
(V.)[n].G.HMSi	显示的 "i" 主轴的 "M" 功能记录	—	—	R	是

变量 GS 和 MS 返回编码值。每种功能有一位表示相关功能是有效 (=1) 或无效 (=0)。状态查询实例：

(V.)G.GS[1] 说明 G1 是有效 (=1) 还是无效 (=0)

(V.)G.MS[6] 说明 M6 是有效 (=1) 还是无效 (=0)

HGS1..10 变量返回 32 位编码状态；每个功能占 1 位。HGS1 变量对应功能 G0 ~ G31，HGS2 对应 G32 ~ G63 等。

HGS 和 HMS 返回编码值； 每个功能有一位表示相关变量是否被显示，显示为 (=1) ，不显示为 (=0)。位 0 为最低有效位，对应 G0 或 M0 功能，位 1 对应 G1 或 M1 等等。

这些变量为 读 / 写 (R/W) 型，在程序段准备期间进行赋值 。

参数和变量		PRG	PLC	INT
(V.)P.name	程序的局部用户变量	R/W	—	—
(V.)S.name	程序的全局用户变量	R/W	—	—
(V.)C.(A-Z)	固定循环调用参数值	R/W	—	—
(V.)C.CALLP(A-Z)	固定循环调用中编写的参数 "0" = 没有编写 "1" = 已经编写	R	—	—
(V.)C.P_(A-Z)	定位循环调用参数值	R/W	—	—
(V.)C.P_CALLP(A-Z)	定位循环调用中编写的参数 "0" = 没有编写 "1" = 已经编写	R	—	—
(V.)C.PCALLP(A-Z)	子程序调用、 G18x、 #PCALL 或 #MCALL 中编写的参数 "0" = 没有编写 "1" = 已经编写	R	—	—

从程序种调用局部和全局子程序时， "(V.)P.name" 变量的值保持不变。

"(V.)S.name" 变量在程序间和复位 （reset）之后的值保持不变。用指令 #DELETE 初始化变量。

```
G90 G81 Z0 I-15
   V.C.CALLP_Z = 1
   V.C.CALLP_I = 1
   V.C.CALLP_K = 0
   V.C.Z = 0
   V.C.Z = -15
```

```
G160 A30 X100 K10 P6
   V.C.P_CALLP_A = 1
   V.C.P_CALLP_K = 1
   V.C.P_CALLP_R = 0
   V.C.P_A = 30
   V.C.P_X = 100
```

```
#PCALL sub.nc A12.56 D3
   V.C.PCALLP_A = 1
   V.C.PCALLP_D = 1
```

这些变量是只读同步型，在程序段准备期间进行赋值。

与圆弧相关的变量		PRG	PLC	INT
(V.)[n].G.R	圆弧半径	R	R	R
(V.)[n].G.I/J/K	圆弧中心坐标 (I, J, K)	R	R	R
(V.)[n].G.CIRERR[i]	圆弧中心校正	R	R	R

下述实例的起点均是 X0 Y0。

功能 G265 有效时，如果圆弧不精确但在公差范围内， CNC 重新计算圆弧中心。

```
G2 X120 Y120.001 I100 J20
V.G.R = 101.980881
V.G.I = 100.0004
V.G.J = 20.0004
V.G.CIRERR[1] = -0.000417
V.G.CIRERR[2] = -0.000417
```

功能 G264 有效时，如果圆弧不精确但在公差范围内，使用从起点计算得出的半径来生成圆弧，保持圆心不变。

```
G2 X120 Y120.001 I100 J20
V.G.R = 101.981371
V.G.I = 100
V.G.J = 20
V.G.CIRERR[1] = 0
V.G.CIRERR[2] = 0
```

这些变量为只读 (R) 同步型，在程序段准备期间进行赋值。

镜像		PRG	PLC	INT
(V.)[n].G.MIRROR	有效镜像	R	R	R
(V.)[n].G.MIRROR1	在通道第一轴有效的镜像	R	R	R
(V.)[n].G.MIRROR2	在通道第二轴有效的镜像	R	R	R
(V.)[n].G.MIRROR3	在通道第三轴有效的镜像	R	R	R

(V.)[n].G.MIRROR 使用最低有效位，每根轴一个(1= 有效 和 0=无效)。最低有效位用于第一轴，下一个用于第二轴，依此类推。

比例缩放因子		PRG	PLC	INT
(V.)[n].G.SCALE	指定有效通用比例缩放因子	R	R	R

极坐标原点		PRG	PLC	INT
(V.)[n].G.PORGF	相对于工件零点的极坐标原点位置 (横坐标)	R	R	R
(V.)[n].G.PORGS	相对于工件零点的极坐标原点位置 (纵坐标)	R	R	R

坐标系旋转 (图形旋转)		PRG	PLC	INT
(V.)[n].G.ROTPF	相对于工件零点的旋转中心位置 (横坐标)	R	R	R
(V.)[n].G.ROTPS	相对于工件零点的旋转中心位置 (纵坐标)	R	R	R
(V.)[n].G.ORGROT	坐标系旋转角度	R	R	R

从动轴		PRG	PLC	INT
(V.)[n].G.LINKACTIVE	从动状态	R	R	R

程序段循环		PRG	PLC	INT
(V.)[n].G.PENDRPT	#RPT 的待循环次数	R	R	R
(V.)[n].G.PENDNR	NR 的待循环次数	R	R	R

(V.)[n].G.PENDRPT 和 (V.)[n].G.PENDNR 表示待执行的循环次数 。第一次执行时，它的值是编写的循环次数减一；最后一次执行时，它的值是零。

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。



HSC 功能		PRG	PLC	INT
(V.)[n].G.HSC	有效 HSC 功能	R	—	—

固定循环		PRG	PLC	INT
(V.)[n].G.CYCLETYPEON	有效固定循环类型	R	R	R

这些变量是同步只读型，在执行程序段时进行赋值。它们对应线性轴和旋转轴。

探测 (G100, G101, G102)		PRG	PLC	INT
(V.)[n].A.MEAS.Xn	实测值 . 刀具基准坐标	R	R	R
(V.)[n].A.ATIPMEAS.Xn	实测值 . 刀尖坐标	R	—	—
(V.)[n].G.PLMEAS1	通道第一轴实测值 . 刀尖坐标	R	—	—
(V.)[n].G.PLMEAS2	通道第二轴实测值 . 刀尖坐标	R	—	—
(V.)[n].G.PLMEAS3	通道第三轴实测值 . 刀尖坐标	R	—	—
(V.)[n].A.MEASOF.Xn	关于编程点的差值	R	R	R
(V.)[n].A.MEASOK.Xn	探测完成 "0" = 否 "1"= 是	R	R	R
(V.)[n].A.MEASIN.Xn	包含测量偏置的坐标	R	R	R
(V.)[n].G.PLMEASOKx	平面轴探测完成	R	—	—

下述实例中，起点位置为 X0，编写了 G100 X100 F100 指令。使用 G101 探测时，(V.)A.MEASIN.Xn 的值被更新。

```
V.A.MEAS.X = 95
V.A.MEASOF.X = -5
V.A.MEASOK.X = 1
```

3D 进给率		PRG	PLC	INT	Exec
(V.)[n].G.F3D	3D 刀具路径的理论进给率	R/W	R/W	R/W	是

这些变量是同步只读 (R) 型， 在程序段准备期间进行赋值。

探针		PRG	PLC	INT
(V.)[n].G.ACTIVPROBE	有效探针数量	R	R	R

这些变量是只读同步型，在执行程序段时进行赋值。这些变量对应线性轴和旋转轴，不对应主轴。

在手动干涉模式下的运动		PRG	PLC	INT
(V.)[n].A.MANOF.Xn	检查或使用 G200 时的距离	R	R	R
(V.)[n].A.ADDMANOF.Xn	使用 G201 时的运动距离	R	R	R

在程序段执行期间即便取消手动干预，这些值仍保持不变。



这些变量可同步读 / 写 (R/W)，在程序段准备期间进行赋值。这些变量对应线性轴和旋转轴。

运动学 (位置)		PRG	PLC	INT
(V.)[n].G.POSROTF	主控旋转轴的当前位置	R/W	R/W	R/W
(V.)[n].G.POSROTS	第二旋转轴的当前位置	R/W	R/W	R/W
(V.)[n].G.TOOLORIF1	主控旋转轴的目标位置	R	R	R
(V.)[n].G.TOOLORIS1	第二旋转轴的目标位置	R	R	R
(V.)[n].G.TOOLORIF2	主控旋转轴的目标位置	R	R	R
(V.)[n].G.TOOLORIS2	第二旋转轴的目标位置	R	R	R

它们指定主轴头的旋转轴所占据以及为了使刀具垂直于定义平面而必须占据的目标位置。当主轴不是全自动时 (单旋转或手动主轴)，它们非常有用。

对于角度 （旋转）主轴头，当计算该目标位置时有两种可以采用的解决方法：

(V.)G.TOOLORIF1 和 (V.)G.TOOLORIS1 指定相对于零点位置主控旋转轴的最短路径。

(V.)G.TOOLORIF2 和 (V.)G.TOOLORIS2 指定相对于零点位置主控旋转轴的最长路径。

这些变量是同步只读 (R) 型，在程序段准备期间进行赋值。它们对应线性轴和旋转轴。

斜面		PRG	PLC	INT
(V.)[n].G.CS	有效 CS 功能编号	R	R	R
(V.)[n].G.ACS	有效 ACS 功能编号	R	R	R
(V.)[n].G.TOOLCOMP	有效补偿功能 "1" = RTCP "2" = TLC "3" = 空	R	R	R

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。

由斜面产生的模具		PRG	PLC	INT
(V.)[n].G.CSMAT1	由斜面产生的模具 . 第 1 行第 1 列	R	R	R
(V.)[n].G.CSMAT2	由斜面产生的模具 . 第 1 行第 2 列	R	R	R
(V.)[n].G.CSMAT3	由斜面产生的模具 . 第 1 行第 3 列	R	R	R
(V.)[n].G.CSMAT4	由斜面产生的模具 . 第 2 行第 1 列	R	R	R
(V.)[n].G.CSMAT5	由斜面产生的模具 . 第 2 行第 2 列	R	R	R
(V.)[n].G.CSMAT6	由斜面产生的模具 . 第 2 行第 3 列	R	R	R
(V.)[n].G.CSMAT7	由斜面产生的模具 . 第 3 行第 1 列	R	R	R
(V.)[n].G.CSMAT8	由斜面产生的模具 . 第 3 行第 2 列	R	R	R
(V.)[n].G.CSMAT9	由斜面产生的模具 . 第 3 行第 3 列	R	R	R
(V.)[n].G.CSMAT10	相对于第一轴机床零点当前坐标系的偏置	R	R	R
(V.)[n].G.CSMAT11	相对于第二轴机床零点当前坐标系的偏置	R	R	R
(V.)[n].G.CSMAT12	相对于第三轴机床零点当前坐标系的偏置	R	R	R

这些变量对应于从理论参考系到实际参考系的转换矩阵。

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。

通道的同步		PRG	PLC	INT
(V.)[n].G.MEETST[i]	[n] 通道中 MEET 型 [i] 标志的状态	R	R	R
(V.)[n].G.WAITST[i]	[n] 通道中 WAIT 型 [i] 标志的状态	R	R	R
(V.)[n].G.MEETCH[i]	[i] 通道的 [n] 通道期望的 MEET 型标志	R	R	R
(V.)[n].G.WAITCH[i]	[i] 通道的 [n] 通道期望的 WAIT 型标志	R	R	R



这些变量是同步只读 (R) 型，在执行程序段时进行赋值。它们对应线性轴、旋转轴以及主轴。

前馈和 和 AC- 前馈		PRG	PLC	INT
(V.)[n].A.FFGAIN.Xn	前馈有效百分率	R	R	R
(V.)[n].A.ACFGAIN.Xn	AC- 前馈有效百分率	R	R	R
(V.)[n].A.ACTFFW.Xn	即时前馈	R	R	R
(V.)[n].A.ACTACF.Xn	即时 AC- 前馈	R	R	R

ACFGAIN 的PLC 读数以十分之一 (x10) 给出。读取 *FFGAIN* 的PLC 以百分之一 (x100) 给出。参考 313 页 *“从 PLC 访问数字值”*。





15.17 与独立轴相关的变量

它们有通用名字。

- 使用通道号替代“n”字符，保留括号。使用编号 1 来识别第一个通道，“0”为无效编号。
- 用轴通道名称、逻辑号或索引替代字符 "Xn" 。

这些变量可同步读 / 写 (R/W) ，在执行程序段时进行赋值。

独立轴		PRG	PLC	INT
(V.)[n].G.IBUSY	执行中的独立轴	R	R	R

这些变量可同步读 / 写 (R/W) ，在执行程序段时进行赋值。它们对应线性轴和旋转轴。

独立轴 (定位)		PRG	PLC	INT
(V.)[n].A.IORG.Xn	独立轴偏置	R/W	R/W	R/W
(V.)[n].A.IPRGF.Xn	独立轴编程进给率	R	R	R
(V.)[n].A.IPPOS.Xn	独立轴编程坐标	R	R	R
(V.)[n].A.ITPOS.Xn	独立轴理论坐标	R	R	R

这些变量可同步读 / 写 (R/W) ，在执行程序段时进行赋值。它们对应线性轴、旋转轴以及主轴。

独立轴 (同步)		PRG	PLC	INT
(V.)[n].A.SYNCTOUT.Xn	建立同步的最大时间	R/W	R/W	R/W
(V.)[n].A.SYNCVEL.Xn	同步速度	R/W	R/W	R/W
(V.)[n].A.SYNCPOSW.Xn	开始校正它时的最大位置差	R/W	R/W	R/W
(V.)[n].A.SYNCVELW.Xn	开始校正它时的最大速度差	R/W	R/W	R/W
(V.)[n].A.SYNCPOSOFF.Xn	同步的位置偏置	R/W	R/W	R/W
(V.)[n].A.SYNCVELOFF.Xn	同步的速度偏置	R/W	R/W	R/W
(V.)[n].A.GEARADJ.Xn	同步期间齿轮传动比微调	R	R	R

GEARADJ 的 PLC 读数以百分之一 (x100) 给出。参考 313 页 *“从 PLC 访问数字值”*。



15.18 机床配置相关的变量

它们有通用名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 用轴或主轴通道中名称、逻辑号或索引替代字符 "Xn" 。
- 使用数字替代 “i” 和 "x" 字符，保留括号。

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。

机床配置		PRG	PLC	INT
(V.)G.NUMCH	通道编号	R	R	R
(V.)[n].G.AXISCH	通道轴名	—	—	R
(V.)[n].A.ACTCH.Xn	轴或主轴的当前通道	R	R	R
(V.)[n].A.ACTIVSET.Xn	有效轴或主轴组	R	R	R
(V.)[n].G.AXIS	通道轴编号	R	R	R
(V.)[n].G.NAXIS	包含屈服轴空位的通道轴编号	R	R	R
(V.)[n].G.AXISNAMEx	通道 "x" 轴的名称	R	R	R
(V.)G.GAXISNAMEx	系统 "x" 轴的名称	R	R	R
(V.)[n].G.NSPDL	通道主轴的编号	R	R	R
(V.)[n].G.SPDLNAMEx	通道 "x" 主轴的名称	R	R	R
(V.)G.GSPDLNAMEx	系统 "x" 主轴的名称	R	R	R
(V.)[n].G.MASTERSP	通道的主要主轴	R	R	R

当搁置轴时，这方法可了解哪根轴可用。变量 (V.)[n].G.AXISNAME 和 (V.).G.GAXISNAME 指定可用的轴。如果一根轴不可用，该变量返回 "?"。

这些变量可同步读 / 写 (R/W) ，它们对应线性轴和旋转轴。

线性轴和旋转轴行程极限		PRG	PLC	INT	Exec
(V.)[n].A.POSLIMIT.Xn	正向软件极限	R/W	R	R	否
(V.)[n].A.NEGLIMIT.Xn	负向软件极限	R/W	R	R	否
(V.)[n].A.RTPOSLIMIT.Xn	第二正向软件行程极限	R/W	R/W	R/W	是
(V.)[n].A.RTNEGLIMIT.Xn	第二负向软件行程极限	R/W	R/W	R/W	是
(V.)[n].G.SOFTLIMIT	达到的软件极限	R	R	R	是

有两个软件极限。CNC 应用最大极限的那个。

变量 POSLIMIT 和 NEGLIMIT 对应机床参数设置的极限。当改变这些变量时，从改变的瞬间起，CNC 采用新值作为新极限 。

它们的值在复位（Reset）后保持不变，但是在确认机床参数和打开 CNC 时被复位。变量 POSLIMIT 和 NEGLIMINT 采用机床参数值， RTPOSLIMIT 和 RTNEGLIMIT 采用最大值。

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。它们对应线性轴和旋转轴。

运动学 (维数)		PRG	PLC	INT
(V.)[n].A.HEADOF.Xn	运动学的维数	R	R	R

它返回该轴有效运动的测量结果。它可以是一个 DATA(运动表格) 的特殊值或由运动类型确定的多个特殊值的组合。



这些变量可同步读取 (R)。

工作平面和轴		PRG	PLC	INT	Exec
(V.)[n].G.PLANE	组成工作平面的轴	R	R	R	否
(V.)[n].G.PLANE1	通道第一主控轴 (横轴)	R	R	R	否
(V.)[n].G.PLANE2	通道第二主控轴 (纵轴)	R	R	R	否
(V.)[n].G.PLANE3	通道第三主控轴	R	R	R	否
(V.)[n].G.PLANELONG	通道垂直轴	R	R	R	否
(V.)[n].G.LONGAX	垂直轴	R	R	R	否
(V.)[n].G.PLAXNAME1	主控轴 (横轴)	—	—	R	是
(V.)[n].G.PLAXNAME2	主控轴 (纵轴)	—	—	R	是
(V.)[n].G.PLAXNAME3	主控轴 (垂直轴)	—	—	R	是
(V.)[n].G.TOOLDIR	刀具定位	R	R	R	否

由 (V.)[n].G.PLANE 和 (V.)[n].G.LONGAX 返回的值按如下方式进行编码：

X=10 X1=11 X2=12 X3=13 ... X9=19

Y=20 Y1=21 Y2=22 Y3=23 ... Y9=29

Z=30 Z1=31 Z2=32 Z3=33 ... Z9=39

U=40 U1=41 U2=42 U3=43 ... U9=49

V=50 V1=51 V2=52 V3=53 ... V9=59

W=60 W1=61 W2=62 W3=63 ... W9=69

A=70 A1=71 A2=72 A3=73 ... A9=79

B=80 B1=81 B2=82 B3=83 ... B9=89

C=90 C1=91 C2=92 C3=93 ... C9=99

因而，如果选择 G17 平面，将获得：

```
V.G.PLANE = 1020          XY 轴 ( 工作平面 )
V.G.LONGAX = 30           Z 轴 ( 垂直 )
G.PLAXNAME1 = X           ( 横坐标轴 )
G.PLAXNAME2 = Y           ( 纵坐标轴 )
G.PLAXNAME3 = Z           ( 垂直坐标轴 )
```

这些变量可同步读 / 写 (R/W) ，在执行程序段时进行赋值。

模拟输入和输出		PRG	PLC	INT
(V.)G.ANAI[i]	[n] 输入电压 (单位伏特)	R	R	R
(V.)G.ANAO[i]	[n] 输出电压 (单位伏特)	R/W	R/W	R

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。它们对应线性轴、旋转轴以及主轴。

反馈输入		PRG	PLC	INT
(V.)[n].A.COUNTER.Xn	反馈脉冲 (整数 + 分数)	R	R	R
(V.)[n].A.COUNTERST.Xn	计数器状态	R	R	R
(V.)[n].A.ASINUS.Xn	A 信号部分	R	R	R
(V.)[n].A.BSINUS.Xn	B 信号部分	R	R	R

对于被激活计数器，必须有一根与它相关的模拟轴。



这些变量可同步读 / 写 (R/W)，在执行程序段时执行。它们对应线性轴、旋转轴以及主轴。

级联轴相关的变量		PRG	PLC	INT
(V.)[n].A.TPIIN.Xn	级联轴的主控轴 PI 输入 (转 / 分)	R	R	R
(V.)[n].A.TPIOUT.Xn	级联轴的主控轴 PI 输出 (转 / 分)	R	R	R
(V.)[n].A.TFILTOUT.Xn	预载过滤器输出	R	R	R
(V.)[n].A.PRELOAD.Xn	预载过滤器输入	R/W	R/W	R/W
(V.)[n].A.FTEO.Xn	SERCOS 的速度指令	R	R	R
(V.)[n].A.TORQUE.Xn	SERCOS 中的当前扭矩	R	R	R

TORQUE 的 PLC 读数以十分之一 (x10) 给出，参考 313 页 *“从 PLC 访问数字值”*。

这些变量可同步读 / 写 (R/W)，在执行程序段时进行赋值。它们对于线性轴、旋转轴以及主轴是有效的。

通过 PLC 设置的变量		PRG	PLC	INT
(V.)[n].A.PLCFFGAIN.Xn	从 PLC 编写的前馈 %	R	R/W	R
(V.)[n].A.PLCACFGAIN.Xn	从 PLC 编写的 AC- 前馈 %	R	R/W	R
(V.)[n].A.PLCPROGAIN.Xn	从 PLC 编写的比例增益	R	R/W	R

为了把以这种方式定义的前馈 AC-前馈考虑在内，必须通过机床参数激活它们，也就是说，如果它是模拟驱动器，使用机床参数 FFWTYPE；如果是 Sercos 驱动器，则使用机床参数 OPMODEP。

由这些变量定义的值优先于由机床参数或编程定义的值。入彀变量值为负，则不起作用 ("0" 是有效值)。这些变量不能通过复位 (reset) 或在确定参数时进行初始化。

PLC 将在下列单元读取它们。参考 313 页 *“从 PLC 访问数字值”*。

PLCACFGAIN 的 PLC 读数以十分之一 (x10) 给出。

从 PLC 设置 Z 轴变量为 ·99.1·：

```
()=MOV 991 R1
()=CNCWR(R1,A.PLCACFGAIN.Z,M1000)
```

PLCFFGAIN 的 PLC 读数以百分之一 (x100) 给出。

从 PLC 设置 X 轴变量为 ·99.12·：

```
()=MOV 9912 R1
()=CNCWR(R1,A.PLCFFGAIN.X,M1000)
```

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。

调整位置的变量		PRG	PLC	INT
(V.)[n].A.POSINC.Xn	当前采样周期的实际位置增量	R	R	R
(V.)[n].A.TPOSINC.Xn	当前采样周期的理论位置增量	R	R	R
(V.)[n].A.PREVPOSINC.Xn	先前采样周期的实际位置增量	R	R	R

微调变量		PRG	PLC	INT
(V.)[n].A.FEED.Xn	实际即时进给率值	R	R	R
(V.)[n].A.TFEED.Xn	理论即时进给率值	R	R	R
(V.)[n].A.ACCEL.Xn	实际即时加速度值	R	R	R
(V.)[n].A.TACCEL.Xn	理论即时加速度值	R	R	R
(V.)[n].A.JERK.Xn	实际即时加加速度值	R	R	R
(V.)[n].A.TJERK.Xn	理论即时加加速度值	R	R	R



15.19 其它变量

它们有通用名字。

- 使用通道号替代 “n” 字符，保留括号。使用编号 1 来识别第一个通道，“0” 为无效编号。
- 用主轴通道中名称、逻辑号或索引替代字符 "Xn" 。
- 使用数字替代 “i” 字符，保留括号。

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。

软件版本		PRG	PLC	INT
(V.)G.VERSION	CNC 版本和版本编号	R	R	R

CNC 状态		PRG	PLC	INT
(V.)[n].G.STATUS	CNC 状态 (摘要)	R	R	R
(V.)[n].G.FULLSTATUS	CNC 状态 (详细)	R	R	R

CNC 状态信息是二进制编码，如下所示：

STATUS

0000	(0H)	未准备
0001	(1H)	准备
0010	(2H)	执行
0100	(4H)	中断
1000	(8H)	错误

FULLSTATUS

高位部分包含 STATUS 变量的信息，低位部分提供更多编码信息。FULLSTATUS = 0000 (STATUS) 0000 (编码)。

对于 FULLSTATUS 的低位部分，其编码列表为：

0000	(0H)	处于复位模式
0001	(1H)	处于 JOG 模式
0010	(2H)	处于 MDI 模式
0011	(3H)	处于编程模式
0100	(4H)	用 M0 停止
0101	(5H)	用 STOP 停止
0110	(6H)	在单独程序段模式停止
1001	(9H)	检查语法
1010	(AH)	程序段搜索 (轴不运动)
1011	(BH)	程序段搜索完成 . 待机模式
1100	(CH)	计算执行时间
1101	(DH)	处于模拟模式

实例：

在 RESET 模式下， FULLSTATUS 的低位是 "0" (0000)；在 JOG 模式下，它的值是 "1" (0001)；在 SIMULATION 模式下是 13 (1101)，等等。

FULLSTATUS=514 (202H) 表示正在执行 (0010) + MDI (0010)。

错误和警告		PRG	PLC	INT
(V.)[n].G.CNCERR	在指定通道中有最高优先级的错误编号	R	R	R
(V.)[n].G.CNCWARNING	在指定通道中显示警告的编号	R	R	R



两个变量使用复位进行初始化。如果几个通道在同一组，通道复位意味着使它们复位，从而初始化该组所有通道的变量。

如果几个通道在同一组，通道中的错误导致在所有通道出现同一错误。在这种情况，该组所有通道的变量 (V.)[n].G.CNCERR 将拥有相同值 。

如果有多个警告，排出它们时变量值被更新。消除最后一个警告后，变量 (V.)[n].G.CNCWARNING 被初始化为零。

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。

时间		PRG	PLC	INT
(V.)G.DATE	日期格式为年 - 月 - 日 (1999 年 4 月 25 日 => 990425)	R	R	R
(V.)G.TIME	时间格式为小时 - 分钟 - 秒 (18 时 22 分 34 秒 => 182234)	R	R	R
(V.)G.CLOCK	从 CNC 启动后经过的时间	R	R	R
(V.)[n].G.CYTIME	零件程序执行时间 （以每秒的百分之一为单位）	R	R	R

(V.)[n].G.CYTIME 在开始新的执行时（即便是相同的程序），它被设置成 0。它不测量已经停止的执行的时间。

这些变量可同步读 / 写 (R/W) ，在执行程序段时进行赋值。

零件计数器		PRG	PLC	INT
(V.)[n].G.PARTC	零件计数器	R/W	R/W	R/W
(V.)[n].G.FIRST	程序第一次执行时	R	R	R/W

(V.)[n].G.PARTC 执行新程序和每次执行 M30 或 M02 时初始化。

(V.)[n].G.FIRST 选择新的程序时，认为是第一次执行 (=1) 。

必须记住，当在通道中改变被执行的程序时 （甚至使用指令 #EXEC），两个变量被初始化。例如，当选择和执行下列程序时，两个变量被初始化。执行指令 #EXEC 时，两个变量被重新初始化，因为执行的程序变化了。如果此时程序再次执行，执行中的程序再次改变，两个变量被更新。

```
G0 X100
#EXEC [“program2.nc”, 1]
M30
```

在这种情况下，记录程序执行的次数，推荐在程序结尾处使用类似计数器的算数参数。



这些变量可同步读 / 写 (R/W)， 在执行程序段时进行赋值。

单独程序段，快进功能，等等		PRG	PLC	INT
(V.)[n].G.SBOUT	激活单独程序段功能	R	R	R
(V.)[n].G.SBLOCK	通过键盘激活被请求的单独程序段	R	R	R/W
(V.)[n].G.BLKSKIP	激活程序段跳行功能 (\ )	R	R	R/W
(V.)[n].G.M01STOP	激活条件停止功能 (M01)	R	R	R/W
(V.)[n].G.RAPID	激活快进功能	R	R	R/W

从键盘 (V.)G.SBLOCK 或 PLC (SBLOCK 标志) 能够激活或取消单独程序段功能。要激活它，仅仅需要设置它们中的一个为高电平 (=1)，但是要取消它时，必须将所有的都设置为低电平 (=0)。

通过 PLC(分别标志 M01STOP, BLKSKIP1 和 MANRAPID) 选择条件停止、程序段跳行和快进功能。

这些变量是同步只读 (R) 型。

与程序相关的变量		PRG	PLC	INT	Exec
(V.)[n].G.FILENAME	执行中的程序名称	—	—	R	是
(V.)[n].G.PRGPATH	执行中的程序路径	—	—	R	是
(V.)[n].G.FILEOFFSET	执行中的行所占的位置	R	R	R	是
(V.)[n].G.BLKN	被执行的最后程序段 (编号) (如果为空，值为 -1)	R	R	R	否
(V.)[n].G.LINEN	被执行的程序行编号	R	R	R	是

(V.)[n].G.FILEOFFSET 指定存在程序第一字符和正在被执行的行之间的字符数量。可以用于强调被执行的行。

这些变量可同步读 / 写 (R/W)， 在执行程序段时进行赋值。

与轴和主轴相关的变量		Lin Rot	Spd	PRG PLC	INT
(V.)[n].A.INPOS.Xn	在适当位置的轴和主轴	是	是	R	R
(V.)[n].A.DIST.Xn	轴或主轴的行程距离	是	是	R/W	R/W
(V.)G.ENDREP	所有轴进行重定位	—	—	R	R
(V.)[n].G.SPDLREP	在刀具检验后，用于重定位主轴的 M 功能	—	—	R	R

这些变量可同步读 / 写 (R/W)， 在执行程序段时执行。

模拟键		PRG	PLC	INT
(V.)G.KEY	CNC 认可的最后键的编码	R	R/W	R

(V.)G.KEY 用于读取被 CNC 认可的最后键，或通过写入期望键的编码，从 PLC 模拟键盘操作。

从 PLC 模拟键盘

```
;R110=0 and R111=1
··· = CNCRD(G.KEY, R100, M102)
```
给寄存器 R100 赋予最后按下的键的编码 。
```
··· = CNCWR(R101, G.KEY, M101)
```
它告诉 CNC 一个键被按下，它的编码由寄存器 R101 指定。



这些变量可同步读 / 写 (R/W) 。

通道		PRG	PLC	INT	Exec
(V.)[n].G.CNCHANNEL	通道号	R	R	R	否
(V.)G.FOCUSCHANNEL	具有有效焦点的通道	R	R/W	R/W	是

这些变量是同步只读 (R) 型，在执行程序段时进行赋值。

JOG 运动		PRG	PLC	INT
(V.)[n].G.INTMAN	允许在 jog 模式下运动	R	R	R

当 jog 模式或 TEACH-IN 模式有效时，在刀具检查期间以及功能 G200 和 G201 有效时，允许进行 Jog 运动。







编程语句与指令

16

有两种类型的高级语言指令：编程指令和流程控制指令。

编程指令

该种指令的定义用符号 "#" 后跟指令名称和相应参数。

它们用于多种操作，例如。

- 显示错误、信息等等。
- 编写相对于机床参考零点 （原点）的运动。
- 执行子程序、程序段和程序。
- 通道同步。
- 连接、屏蔽和交换轴。
- 交换主轴。
- C 轴辅助加工。
- 激活碰撞探测。
- 激活手动干涉。
- 激活高速加工。
- 其他操作。

流程控制指令

该种指令的定义用符号 "$" 后跟指令名称和相应参数。

它们用于设置循环和程序跳转。

16.1 编程语句

16.1.1 显示指令 . 在屏幕上显示错误

中断程序执行并显示指定的错误信息。

用指令 #ERROR 编程，选择要显示的错误编号或错误的文本。

#ERROR

通过选择编号显示错误

显示指定错误的编号和数控系统错误表中相应的文本。如果指定错误编号在数控系统错误表中不存在，不显示任何文本。

编程格式：

```
#ERROR [<编号>]
```

参数	含义
<编号>	错误编号。

错误编号必须是整数，可以用常数、参数或算术表达式定义。当使用局部参数时，必须用 P0-P25 编写。

```
#ERROR [100000]
#ERROR [P100]
#ERROR [P10+34]
```

#ERROR

通过选择文本显示错误

显示指定错误的文本。如果没有定义文本，显示空错误窗口 。

编程格式：

```
#ERROR [<文本>]
```

参数	含义
<编号>	错误文本

必须将错误文本定义在引号之间。某些特殊字符定义如下。

\"	在文本中插入引号。
%%	插入字符 %。

```
#ERROR ["消息"]
#ERROR ["参数\"P100\"错误"]
#ERROR ["P12与P14之差 > 40%%"]
```

16. 编程语句与指令
编程语句

CNC 8070

(SOFT V03.0x)

在错误文本中包含外部值

标示符 %D 或 %d 可以用于在文本中插入外部值 (参数或变量) 。带显示的数据值必须在文本后定义。

```
#ERROR ["Wrong %d value",120]
#ERROR ["Tool %D expired",V.G.TOOL]
#ERROR ["Wrong %D - %D values",18,P21]
```

最多可定义 5 个 %D 或 %d ，但是必须有和标示符一样多的数据值。

16.1.2 显示指令．在屏幕上显示警告

在屏幕上显示指定警告，不中断程序执行。

用指令 #WARNING 编程，选择要显示的错误编号或错误文本。

#WARNING

通过选择编号显示警告

显示指定警告的编号和数控系统警告表中相关的文本。如果指定警告的编号在数控系统警告表中不存在，不显示任何文本。

编程格式：

```
#WARNING [< 编号 >]
```

参数	含义
< 编号 >	警告编号。

警告编号必须是整数，可以用常数、参数或算术表达式定义。当使用局部参数时，必须用 P0-P25 编写。

```
#WARNING [100000]
#WARNING [P100]
#WARNING [P10+34]
```

#WARNING

通过选择文本显示警告

显示指定警告的文本。如果没有定义文本，显示空警告窗口 。

编程格式：

```
#WARNING [< 文本 >]
```

参数	含义
< 编号 >	警告文本。

必须将警告文本定义在引号之间。某些特殊字符定义如下。

\"	在文本中插入引号。
%%	插入字符 %。

```
#WARNING ["Message"]
#WARNING ["Parameter \"P100\" is wrong"]
#WARNING ["Difference between P12 and P14 > 40%%"]
```

在警告文本中包含外部值

标示符 %D 或 %d 可以用于在文本中插入外部值 (参数或变量) 。待显示的数据值必须在文本后定义。

```
#WARNING ["Wrong %d value",120]
#WARNING ["Tool %D expired",V.G.TOOL]
#WARNING ["Wrong %D - %D values",18,P21]
```

最多可定义 5 个 %D 或 %d ， 但是必须有和标示符一样多的数据值。

16.1.3 显示指令 . 在屏幕上显示信息

指定信息出现在屏幕的顶部，并且不中断程序执行。信息保持有效，一直到激活新的信息 (当执行程序结束功能 "M02" 或 "M30" 时不取消)。

要显示的文本用指令 #MSG 编程。

#MSG

显示信息

编程格式：

```
#MSG ["< 文本 >"]
```

参数	含义
< 文本 >	信息文本。

信息的文本必须定义在引号之间。某些特殊字符定义如下。

\\" 在文本中插入引号。

%% 插入字符 % 。

如果没有定义文本，信息从屏幕清除。

```
#MSG ["User message"]
#MSG ["The tool \"T1\" is a finishing tool"]
#MSG ["80%% of feedrate is being used"]
#MSG [""]
```

在错误文本中包含外部值

标示符 %D 或 %d 可以用于在文本中插入外部值 (参数或变量) 。待显示的数据值必须在文本后定义。

```
#MSG ["Part number %D", P2]
#MSG ["The current tool is %D", V.G.TOOL]
#MSG ["Finishing F=%D mm/min. and S=%D RPM", P21, 1200]
```

最多可定义 5 个 %D 或 %d ，但是必须有和标示符一样多的数据值。

16.1.4 显示指令．定义图形区尺寸

#DGWZ 定义图形区

图形区能用指令 #DGWZ 进行定义 (定义图形工作区)。

编程格式：

```
#DGWZ [<Xmin>,<Xmax>,<Ymin>,<Ymax>,<Zmin>,<Zmax>]
```

指令的每个参数对应每根轴的极限。

参数	含义
<Xmin>	X 轴最小极限。
<Xmax>	X 轴最大极限。
<Ymin>	Y 轴最小极限。
<Ymax>	Y 轴最大极限。
<Zmin>	Z 轴最小极限。
<Zmax>	Z 轴最大极限。

两个极限可以为正或负，但是轴的最小极限必须小于最大极限。

定义的新图形区一直保持有效，直到定义另一个图形区、修改图形窗口或关闭数控系统。上电时，数控系统采用定义的缺省图形区。

16.1.5 使能和禁能指令

#ESBLK 单段处理开始

#DSBLK 单段处理结束

指令 #ESBLK 和 #DSBLK 激活和取消单段程序段处理。

当执行指令 #ESBLK 时，数控系统执行下列程序段时将其当作单程序段处理。该单段处理方式保持有效，直到通过执行指令 #DSBLK 进行取消。

```
G01 X20 Y0 F850
G01 X20 Y20
#ESBLK
   (单段方式开始)
G01 X30 Y30
G02 X20 Y40 I-5 J5
G01 X10 Y30
G01 X20 Y20
#DSBLK
   (单段方式结束)
G01 X20 Y0
M30
```

在这种方式下，当在 "单段" 模式执行程序时，#ESBLK 和 #DSBLK 之间的程序段群将被连续执行。换句话说，每个程序段后执行不被中断；他将继续直到遇到指令 #DSBLK 。

#ESTOP 激活 CYCLE STOP 信号

#DSTOP 取消 CYCLE STOP 信号

指令 #ESTOP 和 #DSTOP 激活和取消 CYCLE STOP 信号，无论它来自操作面板或是 PLC。

当执行指令 #DSTOP 时，数控系统激活操作面板的 CYCLE STOP 键，CYCLE STOP 信号来自 PLC。保持有效，直到使用指令 #ESTOP 取消 。

#EFHOLD 激活 FEED-HOLD 信号

#DFHOLD 取消 FEED-HOLD 信号

指令 #EFHOLD 和 #DFHOLD 激活和取消来自 PLC 的 FEED-HOLD 信号。

当执行指令 #DFHOLD 时，数控系统取消来自 PLC 的 FEED-HOLD 输入。它保持无效，直到用指令 #EFHOLD 取消。

16.1.6 相对机床参考零点 （原点）编程

使用该数控系统，运动可以是相对机床参考原点的，并临时取消有效零点偏置和刀具半径、长度补偿。

当相对于机床参考零点运动时，用户编写的功能 G70 或 G71 被忽略。运动以 OEM 设置的单位 （毫米或英寸）执行 （数控系统上电时采用的单位）。

编写的运动不允许是极坐标，或其它类型的变换，例如镜像、坐标 （图形）旋转或比例缩放因子。当功能 #MCS 有效时，不允许使用用于设置新原点的功能 （例如 G92, G54-G59, G158, G30 等）。

#MCS 相对机床零点的运动

该指令可以加到任何包含运动的程序段，以在机床参考系执行运动。

```
G92 X0 Y0
G01 X30 Y30 F850
    ( 原点：工件零点 )
#MCS X30 Y30
    ( 原点：机床零点 ( 原点 ))
G01 X40 Y40
    ( 原点：工件零点 )
M30
```

#MCS ON 激活机床坐标系

#MCS OFF 取消机床坐标系

指令 #MCS ON 和 #MCS OFF 激活和取消机床参考系，编写在两指令之间的运动依据机床参考系执行。

```
G92 X0 Y0
G01 X50 Y50
#MCS ON
    ( 原点：机床零点 ( 原点 ))
G01 ···
G02 ···
G00 ···
#MCS OFF
    ( 原点：工件零点 )
G01 X70 Y70
M30
```

两条指令可以在程序段单独编写。

16.1.7 定义和执行子程序

子程序是一组正确定义的程序段，可以在执行程序的任何位置进行一次或多次调用。

有两种类型的子程序，局部子程序和全局子程序。

- 全局子程序作为独立的程序存储在 CNC 存储器中，可以从其它任何正在执行的程序进行调用。
- 局部子程序作为程序的一部分被定义，仅可以从包含它的程序调用。

主程序（或子程序）可以调用子程序，该子程序还可以调用其它子程序，依此类推，数控系统限制子程序调用嵌套级数最大是 20 级。

局部子程序

必须在程序体之前定义。在同一个程序当中可以定义多个局部子程序。

子程序开始用 "%L< 名称 >" 定义，其中 < 名称 > 处可以使用最多 14 个字符，字符可以是大小写字母和数字 (不允许空格)。

子程序结束用定义 M17, M29 或 #RET。

全局子程序

作为单独的程序定义。用于在 CNC 中存储的名称是子程序名。全局子程序的名称不允许使用圆括号，因为这些字符在局部程序里有特殊含义。

和程序用 M30 结束不同的是，全局子程序必须用 M17, M29 或 #RET 结束。

子程序路径定义

#PATH 定义子程序位置

指令 #PATH 可以用于为查找全局子程序定义预定位置，如下所示。

#PATH ["< 文本 >"]

子程序调用中如果没有定义路径，数控系统首先使用该指令定义的路径查找子程序。

```
#PATH ["C:\Cnc8070\Users\Prg\"]
#PATH ["C:\Cnc8070\Users\"]
```

子程序执行

当指定全路径调用全局子程序时，仅在指定的地址执行查找。如果没有指定路径，用下列指令和地址执行查找：

1. 用指令 #PATH 选择地址。
2. 正在执行的程序地址。
3. 用机床参数 SUBPATH 定义的地址。

LL 调用局部子程序

调用局部子程序。

编程格式：

```
LL <sub>
```

参数	含义
<sub>	子程序名称。

```
LL sub2.nc
```

L 调用全局子程序

调用可以定义全路径的全局子程序。

编程格式：

```
L < 路径 ><sub>
```

参数	含义
< 路径 >	子程序位置。
<sub>	子程序名称。

```
L C:\Cnc8070\Users\Prg\sub1.nc
L C:\Cnc8070\Users\sub2.nc
L Sub3.nc
```

#CALL 调用局部或全局子程序

调用可以定义全路径的子程序 (局部或全局)。

编程格式：

```
#CALL < 路径 ><sub>
```

参数	含义
< 路径 >	子程序位置。
<sub>	子程序名称。

当有两个子程序时，一个局部程序和另一个全局程序，且具有相同的名称，应用如下标准。如果调用时定义了路径，数控系统执行全局子程序，否则，将执行局部子程序。

```
#CALL C:\Cnc8070\Users\Prg\sub1.nc
#CALL C:\Cnc8070\Users\sub2.nc
#CALL Sub3.nc
```

#PCALL 调用初始化参数的局部或全局子程序

调用可能定义了全路径的子程序 （局部或全局）。此类调用允许子程序初始化局部参数。

编程格式：

```
#PCALL <path><sub> P0 P1 P2...
```

参数	含义
<path>	子程序位置。
<sub>	子程序名称。

当有两个子程序时，一个局部程序和一个全局程序，且有相同的名称，应用如下标准。如果调用时定义了路径，数控系统执行全局子程序，否则，将执行局部子程序。

调用参数的值可以用两种方式定义。用参数名 P0, P1, P2, 等等。或用字母 (除了 Ñ) ，以 "A" 等同于 P0 ， "Z" 等同于 P25 的方式。

```
#PCALL C:\Cnc8070\Users\Prg\sub1.nc
#PCALL C:\Cnc8070\Users\sub2.nc A12.3 P10=6
#PCALL Sub3.nc A12.3 F45.3 P10=6
```

在子程序调用中使用局部参数时，除了产生新的嵌套级，它还将为局部参数产生新的嵌套级；在 20 级的子程序嵌套中可以有最多 7 级局部参数嵌套。

#MCALL 调用局部或全局子程序 , 变为模态，并初始化参数

调用可能定义了全路径的子程序 （局部或全局）。此类调用允许初始化子程序的局部参数。

使用此类调用，使子程序变成模态的；也就是在连续的运动中子程序保持有效，子程序在每次运动的结束被重复。用指令 #MDOFF 取消模态子程序。

编程格式：

```
#MCALL < 路径 ><sub> P0 P1 P2...
```

参数	含义
< 路径 >	子程序位置。
<sub>	子程序名称。

当有两个子程序时，一个局部程序和一个全局程序，且有相同的名称，应用如下标准。如果在调用时定义路径，数控系统执行全局子程序，否则，将执行局部子程序。

调用参数的值可以用两种方式定义。用参数名 P0, P1, P2 等等。或用字母 (除了 Ñ) ，以 "A" 等同于 P0 ， "Z" 等同于 P25 的方式。

```
#MCALL C:\Cnc8070\Users\Prg\sub1.nc
#MCALL C:\Cnc8070\Users\sub2.nc A12.3 P10=6
#MCALL Sub3.nc A12.3 F45.3 P10=6
```

在子程序调用中使用局部参数时，除了产生新的嵌套级，它还将为局部参数产生新的嵌套级；在 20 级的子程序嵌套可以有最多 7 级局部参数嵌套。

转变功能成非模态

用指令 #MDOFF 在下列情况中取消模态子程序：

- 执行 M02 或 M30 和 RESET 之后。
- 改变工作平面时。
- 编写探测运动 (G100) 时。
- 修改轴的设置 (#FREE AX, #CALL AX 和 #SET AX) 时。
- 调用另一个子程序 (#PCALL, #CALL, L, LL, G180-189)。
- 激活固定循环。

关于子程序模态字符的注意事项

在编写在子程序内部的运动程序段或关联 T 或 M6 的子程序中，不执行模态子程序。使用 NR=0 编写大量程序段重复时，也不执行。

如果当模态子程序有效时，运动程序段包含大量 NR 不等于 0 的重复，运动和子程序将重复执行 NR 次。

如果一个子程序选作模态，将执行包含指令 #MCALL 的程序段，当前子程序终止模态，新选择的子程序将成为模态的。

#MDOFF 转变功能成非模态

指令 #MDOFF 意味着在该程序段中，使用指令 #MCALL 变成模态的子程序将停止模态状态。

16.1.8 执行程序段和程序的指令

能够从在另一个通道正在执行的程序，执行一个通道的程序段甚至程序。

#EXEC

在指定通道执行程序

用该指令，能够在指定通道执行程序。指定通道程序的执行和在指令 #EXEC 之后的程序段并行 （同时）开始。

如果要执行程序的通道忙碌，将发送相关错误信息。

编程格式：

```
#EXEC [<路径><prg>,<通道>]
```

参数	含义
<路径>	文件位置。
<prg>	待执行的程序。
<通道>	待执行程序段所在通道。

```
#EXEC [PRG1.NC,2]
   (在 2 通道执行指定程序。)
#EXEC [C:\CNC8070\USERS\PRG\EXAMPLE.NC,3]
   (在 3 通道执行指定程序。)
```

编程位置

可以通过书写全路径或不书写全路径的方式定义待执行的程序。当调用指定全路径时，仅在指定地址查找它。如果没有指定路径，用下列指令和地址执行查找：

1. 用指令 #PATH 选择的地址。
2. 执行指令 #EXEC 的程序地址。
3. 用机床参数 SUBPATH 定义的地址。

注意事项

如果没有指定通道或和执行指令 #EXEC 的通道一致，指定的程序将作为子程序被执行。在这种情况，功能 M02 和 M30 将执行所有关联动作（初始化、发送到 PLC 等等），除了结束程序。在执行功能 M02 或 M30 之后，继续执行编写在指令 #EXEC 之后的程序段。

包含指令 #EXEC 的程序可以执行、模拟、语法检查或查找特殊程序段。在所有这些情形下，用指令 #EXEC 调用的程序作为原始程序在相同条件被执行 。

#EXBLK 在指定通道执行程序段

用该指令，能够在指定通道执行程序段。如果要执行程序段的通道忙碌，将发送相关错误信息。执行程序段之后，通道恢复以前的工作模式 。

编程格式：

```
#EXBLK [<程序段>,<通道>]
```

参数	含义
<程序段>	要执行的程序段。
<通道>	可选项。程序段执行的通道。

```
#EXBLK [G01 X100 F550, 2]
   (在 2 通道执行程序段)
#EXBLK [T1 M6]
   (在当前通道执行程序段)
```

如果没有指定通道并从程序执行指令，程序段在其自己的通道执行。如果没有指定通道并且在 MDI 方式执行指令，程序段在当前通道执行。

16.1.9 从动电子轴

两轴可以彼此从动，所以它们中一根 （从动）的运动依靠另一根 （主控）的运动。

能够同时有多个耦合 （从动）轴。

耦合轴用指令 #LINK 激活，指令 #UNLINK 取消。当用一对耦合轴到达程序的结束时，在执行 M02 或 M30 之后取消从动。

关于耦合轴的注意事项

尽管指令 #LINK 允许多根轴系，但必须考虑下列限制 :

- 主要轴 (通道的前三根轴) 不能从动。
- 一对主控 - 从动轴中的两根轴都必须是相同类型 (直线或旋转)。
- 一对轴中的主控轴不能是另一对轴中的从动轴。
- 一根轴不能从动于多于一根主控轴。

同样地，没有取消先前的从动对，新的从动 （耦合）关系不能被激活。

#LINK

激活电子耦合 （从动）轴

该指令定义和激活电子耦合轴。多个耦合可以同时被激活。当执行该指令时，定义为从动的所有轴从属于它们相应主控轴。在这些从动轴，当保持耦合状态时不能编写运动 。

该指令也可以用于定义主控轴和从动轴之间最大许可跟随误差值。

编程格式：

```
#LINK [< 主控 >,< 从动 >,< 误差 >][...]
```

参数	含义
< 主控 >	主控轴。
< 从动 >	从动轴。
<error>	可选项。两根轴跟踪误差的最大许可差值。

编写误差值是可选的；如果没有编写，不执行测试。如果是线性轴，最大误差将以毫米或英寸为单位定义，如果是旋转轴，以度为单位定义。

```
#LINK [X,U][Y,V,0.5]
#LINK [X,U,0.5][Z,W]
#LINK [X,U][Y,V][Z,W]
```

#UNLINK

取消电子耦合 （从动）轴

该指令取消有效的轴从动。

```
#UNLINK
    ( 取消轴耦合 )
```

当用一对耦合轴到达程序的结束时，在执行 M02 或 M30 之后取消从动。

16.1.10轴屏蔽

根据不同的加工类型，一些机床可以有两种不同的配置 (轴和主轴)。为了防止元件出现在某个配置中而导致错误 （驱动、反馈系统等等），数控系统允许暂时屏蔽它们。

例如，交换使用直角坐标系轴和交换标准主轴的机床可以有下列轴设置 :

- 对于标准主轴， X Y Z 轴配置 。
- 对于直角坐标系轴，X Y Z A B 轴配置。

在这种情况下，当用标准轴工作时， A 和 B 轴被暂时屏蔽，以忽略它们的信号。

多根轴和主轴可以同时保持屏蔽，但是它们必须一个接一个的被屏蔽（取消屏蔽）。

用指令 #PARK 屏蔽轴和主轴，用指令 #UNPARK 取消屏蔽。在执行 M02 或 M30 以及 RESET 之后，甚至关闭和重新启动数控系统之后，轴和主轴仍保持屏蔽状态。

轴屏蔽的注意事项

如果轴属于主平面，如果它是有效变换的一部分，或是龙门式轴的主控或从动轴，数控系统不允许屏蔽轴。

关于主轴屏蔽的注意事项

在下列情况，数控系统不允许屏蔽主轴。

- 如果主轴没有停止。
- 如果主轴作为 C 轴工作。
- 如果 G96 或 G63 是有效的，并且是通道的主控主轴。
- 如果G33或G95是有效的，并且是通道的主要主轴或者主轴用于和进给率同步。
- 如果属于一对同步主轴，它必须是主控轴或从动轴。

如果屏蔽主轴之后，仅有一根主轴被留在通道，它将变成新主控轴。如果主轴是非屏蔽的并且是通道的唯一主轴，也被用作新主控主轴。

#PARK 屏蔽轴

该指令用于屏蔽被选轴或主轴。当其中的任意一根被屏蔽时，数控系统认为该轴不再属于机床配置并且不再控制它 (忽略从驱动器和反馈系统等等的信号)。

一旦轴或主轴被屏蔽，局部程序不能涉及它 (运动、速度、 M 功能等等)。

编程格式：

```
#PARK <轴 / 主轴>
```

每个元件 (轴或主轴) 必须被分别屏蔽。然而，不必解除第一元件的屏蔽可以屏蔽第二元件 。

当试图屏蔽一个已经屏蔽的轴或主轴，程序被忽略。

```
#PARK A
    ( 屏蔽 "A" 轴 )
#PARK S2
    ( 屏蔽主轴 "S2")
```


#UNPARK 解除屏蔽轴

该指令用于解除被选轴或主轴的屏蔽。当解除其中的一个屏蔽时，数控系统认为轴属于机床配置并开始控制。

编程格式：

```
#UNPARK <轴 / 主轴>
```

轴必须是一根接一根的被解除屏蔽。

当试图解除一个已经解除屏蔽的轴或主轴的屏蔽时，程序被忽略。

```
#UNPARK A
    ( 解除 "A" 轴屏蔽 )
#UNPARK S
    ( 解除 "S" 主轴屏蔽 )
```

16.1.11 修改通道轴的配置

最初，通过机床参数设置，为每个通道分配轴。当执行程序时，通道可以停止它的轴或请求新轴。这种可能性由机床参数 AXISEXCH 确定，该指令确定轴能否改变通道或者是否这种改变是永久的。

程序结束、复位和上电之后，永久性的改变保持有效。初始配置恢复可以通过确认通用参数并再次启动或者用取消改变的局部程序。



如果为给数控系统上电时，出现校验和误差，同样也会恢复机床参数设置。

确定轴能否变换通道

机床参数 AXISEXCH 可以用下列变量查询。

V.MPA.AXISEXCH.Xn

用轴的名称或逻辑编号替换 "Xn" 。

值	含义
0	不能改变通道。
1	临时改变。
2	永久改变。

确定轴所在通道

通过使用下列变量用能够确定轴在哪个的通道。

V.[n].A.ACTCH.Xn

用轴的名称或逻辑编号替换字母 "Xn" 。

用通道编号替换字母 "n" 。

值	含义
0	不在任何通道。
1-4	通道编号。

通过程序修改轴配置的命令

下列指令用于修改轴的配置。能够添加或移除轴，改变轴的名称甚至通过交换名称重新定义通道的主要轴。

改变轴的配置可以取消激活的极坐标原点、图形旋转、镜像和比例缩放因子。

在轴的配置中 (如果 G17 有效)，占第一位的轴必须是横坐标轴，第二是纵坐标轴。第三是垂直于工作平面的轴 ，第四是第一辅助轴等等。

#SET AX

设置轴的配置

在通道定义新的轴配置。在指令中没有编写的通道轴和编写的不存在的轴将被增加。轴放置在和用指令 #SET AX 编写的一样的位置。根据需要，一个或多个偏置可以应用在定义的轴上。

和编写所有轴的 #FREE AX 然后所有新轴的 #CALL AX 一样。

指令 #SET AX 也可以用于单独调用在通道存在的轴。

编程格式：

```
#SET AX [<Xn>,...] <偏置> <...>
```

参数	含义
<Xn>	组成新配置的轴。如果代替定义轴而编写零，在此位置出现一个空的空间 （没有轴）。
< 偏置 >	可选项。设置应用偏置的轴。可以应用多个偏置。

```
#SET AX [X,Y,Z]
#SET AX [X,Y,V1,0,A]
```

偏置设置

用下列指令确定可以应用在轴上的偏置。要应用几个偏置，编写相关命令，并用空格隔开。

命令	含义
ALL	包含所有偏置。
LOCOF	包含参考搜索的所有偏置。
FIXOF	包含夹具偏置。
TOOLOF	包含刀具偏置。
ORGOF	包含零点偏置。
MEASOF	包含测量偏置。
MANOF	包含手动操作偏置。

```
#SET AX [X,Y,Z] ALL
#SET AX [X,Y,V1,0,A] ORGOF TOOLOF
```

如果在定义新的配置时，仅通道轴的次序可以交换时，偏置被忽略。

屏幕显示

首先，轴以在通用参数表（通道）中定义的顺序出现，然后以定义的交换顺序出现。

Y 00000.0000 ? 00000.0000 ? 00000.0000 Z 00000.0000 A 00000.0000	X 00125.1500 Y 00089.5680 Z 00000.0000 ? 00000.0000 ? 00000.0000
#SET AX [Y, 0, 0, Z, A]	#SET AX [X, Y, Z] FIXOF ORGOF

不同配置的屏幕显示，假定机床有五轴 X-Y-Z-A-W。

#CALL AX

给配置增加轴

添加一根或多根轴到预置配置，并允许定义轴的位置。如果在配置中轴已经存在，轴被放在新位置。根据需要，一个或多个偏置可以应用于定义的轴。

编程格式：

```
#CALL AX [<Xn>,<pos>...] <偏置> <...>
```

参数	含义
<Xn>	加给配置的轴。如果轴已经存在，放置到新位置。
<pos>	可选项。在新配置中的轴位置。如果没有编写，轴放置在最后的轴之后。如果位置被占，发送相关错误信息报告。
<offset>	可选项。设置应用于轴的偏置。可以应用多个偏置。

```
#CALL AX [X,A]
    (在已存在的最后一根轴之后，给配置添加 X 和 A 轴)
#CALL AX [V,4,C]
    (添加 V 轴到位置 4，添加 C 轴到最后一根轴之后)
```

偏置设置

可以应用在轴上的偏置和下列指令联系在一起。要应用多个偏置编写相应指令，用空格分隔开。

指令	含义
ALL	包含所有偏置。
LOCOF	包含参考搜索的偏置。
FIXOF	包含夹具偏置。
TOOLOF	包含刀具偏置。
ORGOF	包含零点偏置。
MEASOF	包含测量偏置。
MANOF	包含手动操作偏置。

```
#CALL AX [X] ALL
#CALL AX [V1,4,Y] ORGOF TOOLOF
```

屏幕显示

首先，轴以在通用参数表（通道）中定义的顺序出现，然后以定义的交换顺序出现。

Y 00000.0000 X 00000.0000 W00000.0000 Z 00000.0000 ? 00000.0000	轴设置 #SET AX [Y, 0, 0, Z] Y: 横坐标轴 Z: 第一附加轴 #CALL AX [X,2, W, 3] Y: 横坐标轴 X: 纵坐标轴 W: 垂直于平面的轴 Z: 第一附加轴

#FREE AX

从设置移除轴

从当前设置移除轴。在移除轴之后，位置空出。但是通道中保持轴的顺序不变。

编程格式：

```
#FREE AX [<Xn>,...]
```

参数	含义
<Xn>	从设置中除去的轴。

```
#FREE AX [X,A]
    ( 从设置中移除 X 和 A 轴 )
#FREE AX ALL
    ( 从通道移除所有的轴 )
```

屏幕显示

首先，轴以在通用参数表（通道）中定义的顺序出现，然后以定义的交换顺序出现。

X 00000.0000 Y 00000.0000 Z 00000.0000 A 00000.0000 B 00000.0000	X 00000.0000 ? 00000.0000 Z 00000.0000 ? 00000.0000 B 00000.0000

#FREE AX [Y, A]

不同配置的屏幕显示，假定机床有五轴 X-Y-Z-A-W。

#RENAME AX 重命名轴

改变轴的名称。对于每个编程轴对，第一轴获得第二轴的名称。如果第二轴出现在配置中，它获得第一轴的名称 。

改变的轴名称仅在执行该程序期间保持有效。当开始下一个程序时恢复轴的初始名称。

编程格式：

```
#RENAME AX [<Xn1>,<Xn2>][...]
```

参数	含义
<Xn1>	改变名称的轴。
<Xn2>	新的轴名称。

```
#RENAME AX [X,X1]
  ( X 轴现在叫做 X1。如果 X1 在通道已经存在， 它被叫做 X。 )
#RENAME AX [X1,Y][Z,V2]
```

16.1.12 修改通道主轴的配置

CNC 最多可以有四根主轴，分布在系统的多个通道。一个通道可以有一根或多根主轴，或没有相关主轴。

最初，每个通道有一些赋予它的主轴，与机床参数设置的一样。当执行程序时，通道可以移除它的主轴或者请求新主轴。该可能性由机床参数 AXISEXCH 决定，它确定哪根主轴能改变通道，或者改变是永久的或不是。

在程序结束、复位和上电之后永久改变保持有效。通过确认通用参数和重新启动或者取消变化的局部程序，初始配置可以恢复。



当数控系统上电时，如果出现检查和误差，也会恢复机床参数设置。

确认是否主轴能变换通道

机床参数 AXISEXCH 可用下列变量来查询。

```
V.MPA.AXISEXCH.Sn
```

用主轴名称替换 "Sn" 。

值	含义
0	不能变换通道。
1	临时变换。
2	永久变换。

确认主轴所在通道

通过使用下列变量能够知道主轴在哪个通道。

```
V.[n].A.ACTCH.Sn
```

用主轴名称替换 "Sn" 。

用通道编号替换字母 "n" 。

值	含义
0	不在任何通道。
1-4	通道编号。

通过编程修改主轴配置的指令

下列指令用于修改通道主轴的配置。能够增加或移除主轴，改变主轴的名称和定义通道的主控主轴 。

#FREE SP 从配置释放主轴

从当前配置移除主轴。

编程格式：

```
#FREE SP [<Sn>,...]
#FREE SP ALL
```

参数	含义
<Sn>	主轴名称。
ALL	释放通道所有主轴。

```
#FREE SP [S]
```
(从配置移除主轴 S)

```
#FREE SP [S1,S4]
```
(从配置移除主轴 S1 和 S4)

```
#FREE SP ALL
```
(从配置移除所有主轴)

#CALL SP 给配置添加主轴

给当前配置添加一根或多根主轴。在通道上主轴的位置不是相关的。要添加一根主轴到通道，主轴必须是自由的；必须不在在其它通道。

编程格式：

```
#CALL SP [<Sn>,...]
```

参数	含义
<Sn>	主轴名称。

```
#CALL SP [S1]
```
(添加主轴 S1 到配置)

```
#CALL SP [S,S2]
```
(添加主轴 S 和 S2 到配置)

#SET SP 设置主轴配置

定义新主轴配置。在通道存在的主轴和在 #SET SP 没有编写的主轴被移除，那些编写的没有已经在通道的主轴将被添加。

和编写所有主轴的 #FREE SP 然后所有新主轴的 #CALL SP 一样。编程格式是：

```
#SET SP [<Sn>,...]
```

参数	含义
<Sn>	主轴名称。

```
#SET SP [S]
```
(设置一根主轴)

```
#SET SP [S1,S2]
```
(设置两根主轴)

#RENAME SP 重命名主轴

改变主轴名称。对每个编程主轴对，第一主轴获得第二主轴的名称。如果第二主轴出现在配置中，它获得第一主轴的名称。

改变的主轴名称仅在执行该程序期间保持有效。当开始下一个程序时恢复主轴的初始名称。

编程格式：

```
#RENAME SP [<Sn>,<Sn>][...]
```

参数	含义
<Sn>	主轴名称。

```
#RENAME SP [S,S1]
#RENAME SP [S1,S2][S3,S]
```

16.1.13 选择通道的主控主轴

主控主轴是通道的主要主轴。它是当没有涉及特殊主轴时接受指令的主轴 。

在数控系统上电时和复位之后，用通道机床参数定义的第一主轴作为主控主轴 （初始主控） 。如果该主轴被屏蔽或分配给另一个通道，用通道机床参数定义的下一根主轴等作为主控主轴。如果因为它们被屏蔽或分配给另一个通道，通道没有初始配置 (用机床参数定义的)，没有被屏蔽的当前配置的第一主轴 被用作主控主轴。

当执行 M30 时，遵循相同标准，但是考虑执行该指令之后临时主轴交换没被取消，在下一个程序开始时被取消。意思是执行 M30 之后，初始主控主轴不可以是有效的，但是在下一个程序开始将是有效的。在这种情形下，在 M30 之后，通道将临时采用一个在下一个程序开始时会改变的主控主轴 。

#MASTER

建立通道主控主轴

编程格式：

```
#MASTER <Sn>
```

参数	含义
<Sn>	主轴名称。

```
#MASTER S
#MASTER S2
```

如果没有确定主控主轴，在屏蔽或交换主轴之后，根据下列标准采取主轴主轴。通常，每当通道有单根主轴，它将是主控主轴。

- 如果整个系统仅有一根主轴，它将是当前通道的主控主轴。
- 如果给通道添加一个它没有的主轴，这根主轴将是主控主轴。
- 如果通道释放它的主控主轴，并且仅剩下一根主轴，这根主轴将是新主控主轴。
- 如果通道有两根主轴但是没有主控主轴，通道释放其中的一根主轴，留下的一根主轴将是主控主轴。
- 首先，在有多根主轴的通道，主控主轴将是机床参数设定的主轴。
- 如果通道保持两个或更多主轴并且没有以前的标准可以被应用，则应用下列标准。

  如果任一主轴是初始主控主轴，则采用作主控主轴。如果它被屏蔽了，从初始配置 （用机床参数定义的）选择下一根主轴，依此类推。

  如果从初始设置没有主轴在通道是可用的，则采取当前配置的第一主轴作为主控主轴。如果它被屏蔽，选择下一根主轴，依此类推。

介绍的关于添加或移除主轴的相同方法可应用到屏蔽和取消屏蔽主轴。

16.1.14 主轴同步

该方式可以用于设置以给定比例和另一主轴（主控）同步的主轴（从动）运动。主轴同步总是编写在从动主轴所属的通道，激活、取消或复位它。

有两种同步：速度同步和主轴同步。不同类型同步的激活和取消用下列指令编程。

#SYNC - 基于实际坐标的主轴同步。

#TSYNC - 基于理论坐标的主轴同步。

#UNSYNC - 主轴同步取消。

#SYNC 基于实际坐标的主轴同步

#TSYNC 基于理论坐标的主轴同步

各条指令的编程格式如下。符号 <> 之间为可选参数。

```
#SYNC [master, slave <,nratio> <,dratio> <,posync>
<,synctype>][··]
#TSYNC [master, slave <,nratio> <,dratio> <,posync>
<,synctype>][··]
```

每对括号内的内容定义两根主轴的之间的同步。

参数	含义
slave	同步的从动主轴。
master	同步的主控主轴。
nratio dratio	可选参数。这对数定义同步主轴之间的齿数比。 两个值可以是正的或负的。
posync	可选参数。该参数确定 在适当位置两主轴同步，它还设置在两主轴间的偏置（换档）。 值可以是正的或负的，并且大于 360º。
synctype	可选参数。该参数确定主控主轴的环类型。 "CLOOP" 主轴闭环。"OLOOP" 主轴开环。 如果没有编写，采用 "CLOOP"。

```
#SYNC [S,S1]
```
主轴速度同步。从动主轴以和主控主轴 S 相同的速度旋转。

```
#SYNC [S,S1,1,2]
```
从动主轴 S1 以主控主轴 S 速度的二分之一旋转。

```
#SYNC [S,S1,1,2,0]
```
速度和位置同步之后，从动主轴 S1 以确定的偏置跟随主控主轴，以 0º 偏置为例。

```
#SYNC [S,S1,1,1,30,OLOOP]
```
以 30º 的偏置两主轴速度和位置同步。从动轴以开环方式工作。

同步注意事项

功能 #SYNC 可以在开环 (M3 or M4) 或闭环 (M19) 执行。

可以用相同指令 #SYNC 或 #TSYNC 对多对同步主轴进行编程。只要不和以前的冲突，对于附加作用还可以编写多个连续指令 #SYNC。

从动主轴必须在同步被激活的通道，然而主控主轴可以在任何通道。多个从动主轴可以有相同的主控主轴，但是从动主轴不能做第三根主轴的主控主轴；从而保护同步回路。

编程能够先速度同步然后位置同步或者同时进行。一旦一对主轴已经同步，能够改变速。

为确保正确的后续动作，两根主轴必须在闭环操作。一旦两主轴都在闭环，从动主轴从当前速度转换到同步速度。当编写同步时主控主轴可以是旋转的，它将保持旋转转换到闭环。

编写主控和从动主轴

对于从动主轴，不能够编写速度、主轴功能 M3, M4, M5 和 M19、换档从 M41 到 M44 或改变速度倍率。

下列功能可以对主控主轴编写：

- 通过 DNC, PLC 或 CNC 改变主轴速度。
- 执行速度功能 G94, G95, G96 和 G97。
- 执行辅助功能 M3, M4, M5 和 M19。
- 通过 DNC、 PLC、 CNC 或键盘改变主轴速度倍率。
- 通过 DNC, PLC 或 CNC 改变主轴速度极限。
- 如果 C 轴被激活，定义 XC 平面或 ZC 平面。

当定义同步或主控主轴有效时，主控主轴可以作为 C 轴工作或在指令 G63 方式。功能 G33、 G95 或 G96 也可以在主控主轴是有效的。功能 G33 和 G95 在从动主轴也可以是有效的，但是功能 G96 将保持临时 " 冻结 " 和在同步期间无效。

另一方面，同步主轴不能变换通道，不能执行换档从指令 M41 到 M44。如果换档是自动的，新速度需要换档，将发送相关错误信息 。

工作齿轮

主轴可以有不同的齿轮（范围），如果在同步时，主轴不是相同状态，从动主轴“冻结”状态，更换成用机床参数 SYNCSET 确定的齿轮并且强制跟随主控主轴。

如果主控主轴属于同一通道，也更换成用机床参数 SYNCSET 确定的齿轮。如果主控主轴在另一个通道，在激活同步之前必须激活齿轮。因此，直到用户准备好主控主轴，从动主轴才能和它同步。

原点搜索

如果以前已经回过原点，从动主轴在激活位置同步之前必须进行回原点 （参考）操作 。如果主控主轴在同一通道，也强制回原点。如果主控主轴在另一通道并且没有回过原点，将会发送错误信息报告。

#UNSYNC 解除一根或多根主轴的同步

编程格式：可选参数确定在 <> 之间。

```
#UNSYNC
#UNSYNC [slave1 <,slave2> ...]
```

如果没有定义参数，所有主轴解除同步。

参数	含义
slave	要同步的从动主轴。

```
#UNSYNC
```
通道的所有主轴解除同步。

```
#UNSYNC [S1,S2]
```
从动主轴 S1 和 S2 从它们所同步的主轴解除同步。

解除同步注意事项

也可以用 M30 和 RESET （复位）取消同步。

当取消同步时，主控主轴保持当前状态，从动轴停止。从动轴不恢复同步前的 M 功能，但是保持同步齿轮直到编写新的 S 功能。

和同步运动关联的变量

这些变量是读 / 写 (R/W) 同步，在执行期间计算。它们有通用名。

- 用通道编号替换字符 "n"，保留括号。第一通道用编号1识别，"0"不是有效编号。
- 用名称、逻辑编号或轴的通道指针替换字符 "Xn"。

调整同步速度比

(V.)[n].A.GEARADJ.Xn

从 PRG, PLC 和 INT 只读。

PLC 读数以百分之一 (x100) 给出。

在同步期间精确的调整齿轮比。它编写为初始调整值的百分比。

速度同步

(V.)[n].A.SYNCVELW.Xn

从 PRG, PLC 和 INT 只读。

当主轴是速度同步时，从动主轴以和主控主轴相同的速度旋转 （考虑比率）。如果该变量定义的值被超越， SYNSPEED 信号变低；运动不停止，不发送错误信息报告。

缺省值是机床参数 DSYNCVELW 的值。

(V.)[n].A.SYNCVELOFF.Xn

从 PRG, PLC 和 INT 只读。

从动主轴同步的速度偏置。

位置同步

(V.)[n].A.SYNCPOSW.Xn

从 PRG, PLC 和 INT 只读。

对主轴是位置同步，从动主轴跟随主轴，保持编程偏置 （考虑比率）。如果该变量定义的值被超越， SYNCPOSI 信号变低；运动不停止，不发送错误信号。

缺省值是机床参数 DSYNCPOSW 的值。

(V.)[n].A.SYNCPOSOFF.Xn

从 PRG, PLC 和 INT 只读。

位置偏置。

16.1.15 为轴或主轴选择开环或闭环



该功能对 Sercos 位置驱动（轴或主轴）不可用。在这种情况下，数控系统不能开或关回路，改为驱动控制回路。

当工作在开环时，指令不随反馈而定。当工作在闭环时，指令的产生取决于反馈。

在 M3 或 M4 方式下时，主轴通常工作在开环模式，在 M19 方式下时，主轴通常工作在闭环模式。当主轴同步时，主控主轴和从动主轴总是工作在闭环。可是，进行下列对主轴的调整，能够使用 M3 和 M4 工作在闭环模式：

- 为 M19 调整回路。
- 为当主轴是同步的主控主轴时调整回路。

轴通常工作在闭环。可以在开环控制旋转轴，好像它是主轴一样。

下列指令可以用于开放和闭合回路，它们对轴和对主轴一样是有效的。

`#SERVO ON` - 激活闭环模式。

`#SERVO OFF` - 激活开环模式。

#SERVO ON 激活闭环模式

编写该指令转换轴或主轴到闭环模式。

在转换到闭环模式之前主轴必须回零；否则回路关闭，发送报警信息。

编程格式：

```
#SERVO ON [ 轴 / 主轴 ]
```

参数	含义
轴 / 主轴	轴或主轴名称。

每根轴或主轴的回路必须分别关闭。

```
#SERVO ON [S]
    关闭 S 主轴的回路。
#SERVO ON [S2]
    关闭 S2 主轴的回路。
#SERVO ON [X]
    关闭 X 轴的回路。
```

#SERVO OFF 激活开放回路模式

编写该指令转换轴或主轴到开放模式。

对主轴，编写 #SERVO ON 取消闭环模式，并且恢复关闭回路前的主轴状态。

- 如果主轴在 M19 模式或在主轴同步，在编写该指令之后主轴保持闭环。
- 如果主轴无同步在 M3, M4 或 M5，数控系统开放回路。

编程格式：

```
#SERVO OFF [ 轴 / 主轴 ]
```

参数	含义
轴 / 主轴	轴或主轴的名称。

每根轴或主轴的回路必须分别开放。

```
#SERVO OFF [S]
```
取消主轴 S 的闭合回路。

```
#SERVO OFF [Z2]
```
Z2 轴转换到开放回路模式。

设计回路注意事项

功能 M19 总是包含闭环工作。缺省功能 M3, M4 和 M5 工作在开环，但是当编写主轴同步或指令 #SERVO ON 时，它们也能工作在闭环。

当主轴成为 C 轴或和其余的轴插补 （例如刚性攻螺纹），不失去它拥有的开环或闭环状态。当使用这些指令时，恢复先前的状态。

上电时，主轴设置为开环。执行 M30 或 RESET （复位）之后，数控系统开放回路并且取消指令 #SERVO ON，除了当为同步的主控主轴 （可以在和从动主轴不同的通道）的复位，在这种情况，同步不被取消，不转换到开环。这种情形下会发送报警信息。

16.1.16 垂直刀具轴选择

可以用指令 #TOOL AX 选择刀具的垂直轴。

#TOOL AX

垂直轴选择

该指令允许选择任何机械轴作新垂直轴。

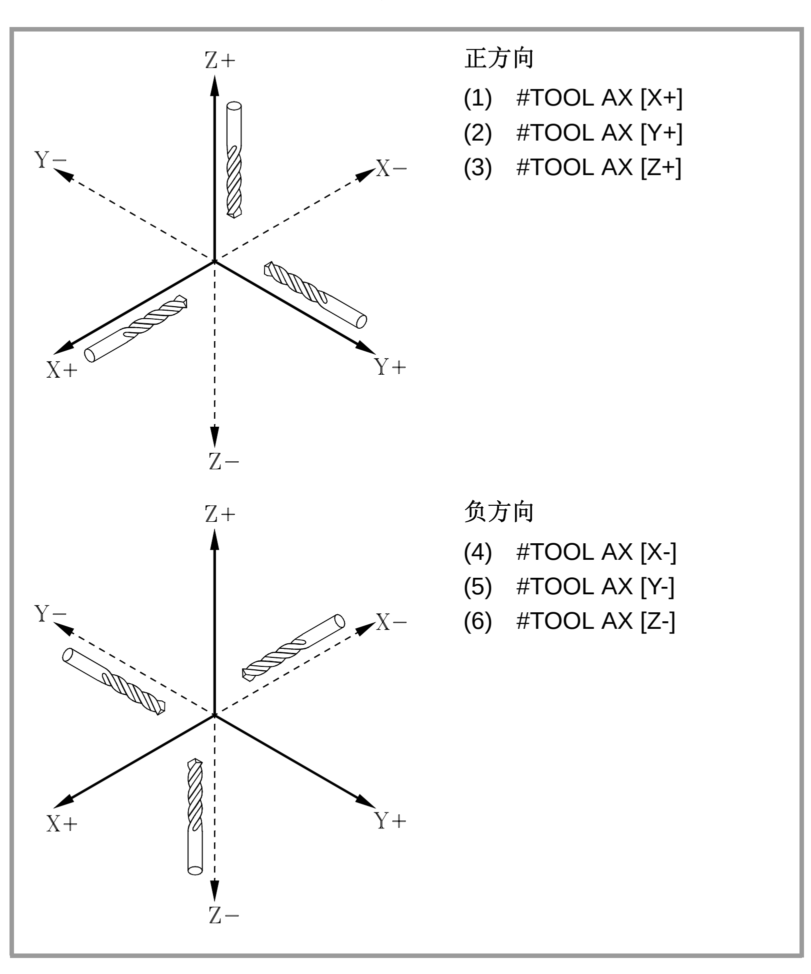
编程格式：

```
#TOOL AX [<axis><+/->]
```

参数	含义
<axis>	刀具的垂直轴。
<+/->	刀具方向。

刀具方向如下确定。

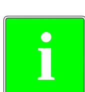
+ 在轴的正方向的刀具位置。
- 在轴的负方向的刀具位置。



.

16.1.17 "C" 轴 : 激活主轴作 "C" 轴

用指令 #CAX and #CAX OFF 可以激活或取消主轴作 "C" 轴 。



要激活轴或主轴作 "C" 轴，必须由机床制造商同样地定义过。
虽然机床可以有多根主轴定义为 "C" 轴，但是仅其中的一根是有效的。

#CAX

激活主轴作 "C" 轴

激活主轴作 "C" 轴。类似旋转轴编写 "C" 轴 （单位度）。

编程格式：

```
#CAX [<Sn>,<名称>]
```

参数	含义
<Sn>	可选参数。激活作 C 轴的主轴。
<名称>	可选参数。C 轴的名称。

当主轴 （除了主控主轴）激活为 C 轴时，主轴需要被唯一地确定。否则，没有必要编写它。

<名称> 参数设置标识 C 轴的名称。该名称将用在局部程序定义运动。如果没有编写，在机床参数 (CAXISNAME) 有缺省名称来命名它。

激活主控主轴作 "C" 轴。

```
#CAX
G01 Z50 C100 F100
G01 X20 C20 A50
#CAX OFF
```

如果多根主轴可以被激活作 C 轴。

```
#CAX [S1,C1]
    (主轴 "S1" 激活为 "C" 轴，名称为 "C1")
G01 Z50 C1=100 F100
G01 X20 C1=20 A50 S1000
#CAX OFF
```

使用 C 轴的注意事项

激活正在运转的主轴作 C 轴将会停止主轴。

当激活主轴作 "C" 轴时，不能对它编写速度。

当激活主轴作 "C" 轴时，数控系统对 "C" 轴执行回原点搜索。

#CAX OFF

取消 C 轴

取消 C 轴，主轴返回作普通主轴。

编程格式：

```
#CAX OFF
```

16.1.18 "C" 轴：零件表面加工

用指令 #FACE 和 #FACE OFF 激活和取消表面加工。对于这种类型的加工，旋转轴或主轴可以用作 "C" 轴。

- 当使用轴时，在用指令 #FACE 定义平面之后激活作 "C" 轴。
- 当使用主轴时，必须在使用指令 #CAX 之前激活作 "C" 轴。



要激活轴或主轴作 "C" 轴，必须由机床制造商同样地定义过 (CAXIS)。根据机床设置，可能需要定义相关运动学 (类型 41/42)。

尽管机床可以有多根轴定义作 "C" 轴，但仅其中的一根是有效的。

#FACE

激活零点回转面的加工

激活零件表面加工并且定义工作平面。激活作 "C" 轴的轴将由定义的工作平面来决定 。

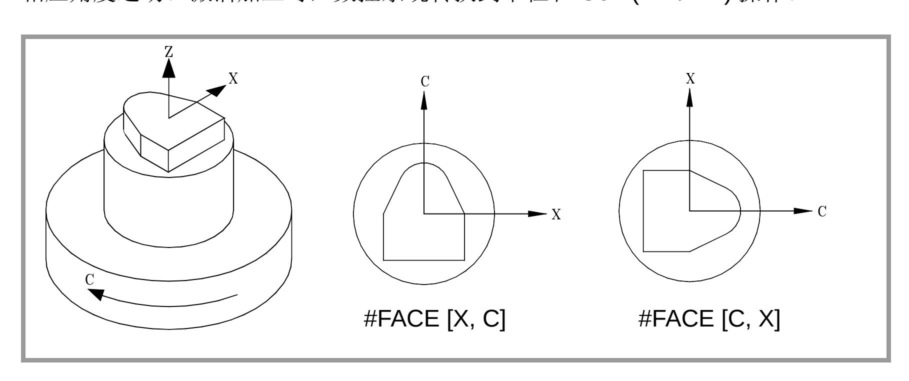
编程格式：

```
#FACE [<abs>,<ord>,<long>]<cin>
```

参数	含义
<abs>	工作平面的横坐标。
<ord>	工作平面的纵坐标。
<long>	可选参数。刀具的垂直轴。
<cin>	可选参数。运动学编号。

编写运动学是可选的。如果没有编写，数控系统应用在机床参数中定义的并且对这种加工类型有效的第一运动学。

"C" 轴编程，好像它是线性轴一样（单位毫米或英寸），数控系统根据所选半径计算相应角度运动。激活加工时，数控系统转换到半径和 G94 (mm/min) 操作。

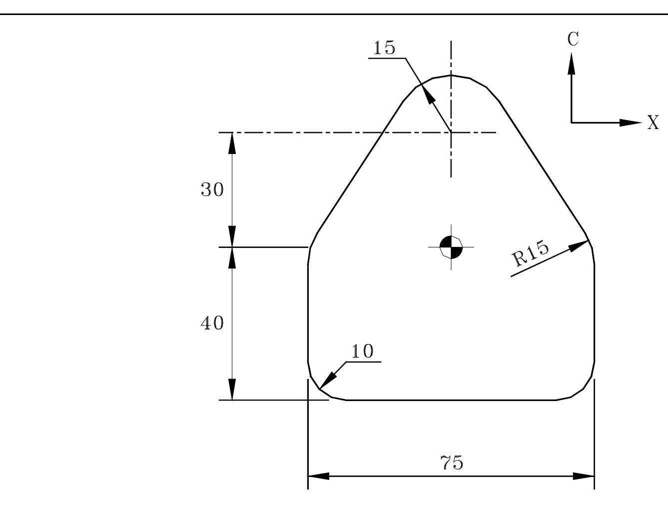


#FACE OFF　　取消零件表面加工

取消零件表面加工。

编程格式：

```
#FACE OFF
```




```
#FACE [X,C]
G90 X0 C-90
G01 G42 C-40 F600
G37 I10
X37.5
G36 I10
C0
G36 I15
X12.56 C38.2
G03 X-12.58 C38.2 R15
G01 X-37.5 C0
G36 I15
C-40
G36 I10
X0
G38 I10
G40 C-90
#FACE OFF
M30
```

16.1.19 "C" 轴 : 零件回转面加工

用指令 #CYL 和 #CYL OFF 激活和取消在回转面加工。对于这种类型的加工，旋转轴或主轴可以用作 "C" 轴。

- 使用轴时，在用指令 #CYL 定义平面之后激活作 "C" 轴。
- 使用主轴时，必须在使用指令 #CAX 之前激活作 "C" 轴。



要激活轴或主轴作 "C" 轴，必须由机床制造商同样地定义过 (CAXIS)。根据机床设置，可能需要定义相关运动学 (类型 43)。

尽管机床可以有多根轴定义作 "C" 轴，但仅其中的一根是有效的。

#CYL

激活零件的回转面加工

激活零件的回转面加工和定义工作平面。激活作 "C" 轴的轴由定义的工作平面确定。

编程格式：

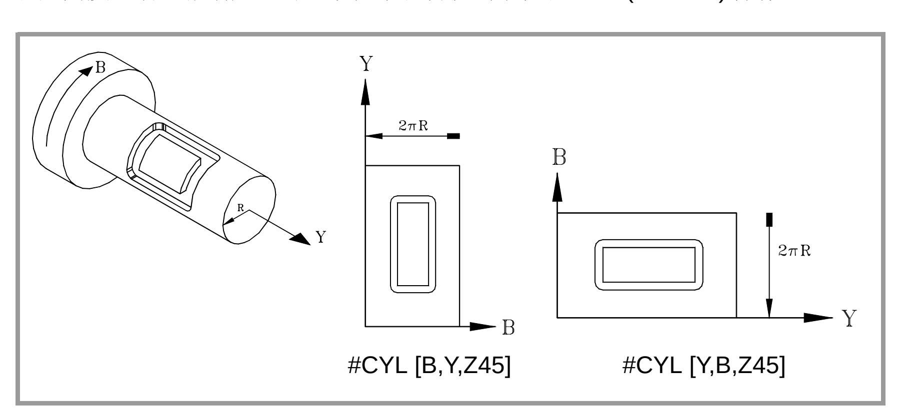
```
#CYL [<abs>,<ord>,<long><radius>]<cin>
```

参数	含义
<abs>	工作平面的横坐标。
<ord>	工作平面的纵坐标。
<long>	刀具的垂直轴。
<radius>	可选参数。将被加工的圆柱体半径 。
<cin>	可选参数。运动学编号 。

编写半径是可选的。如果没有编写，采用从旋转中心到刀尖的距离作为圆柱体半径。使能够在不必指定半径下，在圆柱体表面进行变半径加工。

编写运动学是可选的。如果没有编写， 数控系统应用在机床参数中被定义的并且对这种加工类型有效的第一运动学。

"C" 轴编程，好像它是线性轴一样（单位毫米或英寸），数控系统根据所选半径计算相应角度运动。激活加工时，数控系统转换到半径和 G94 (mm/min) 操作。

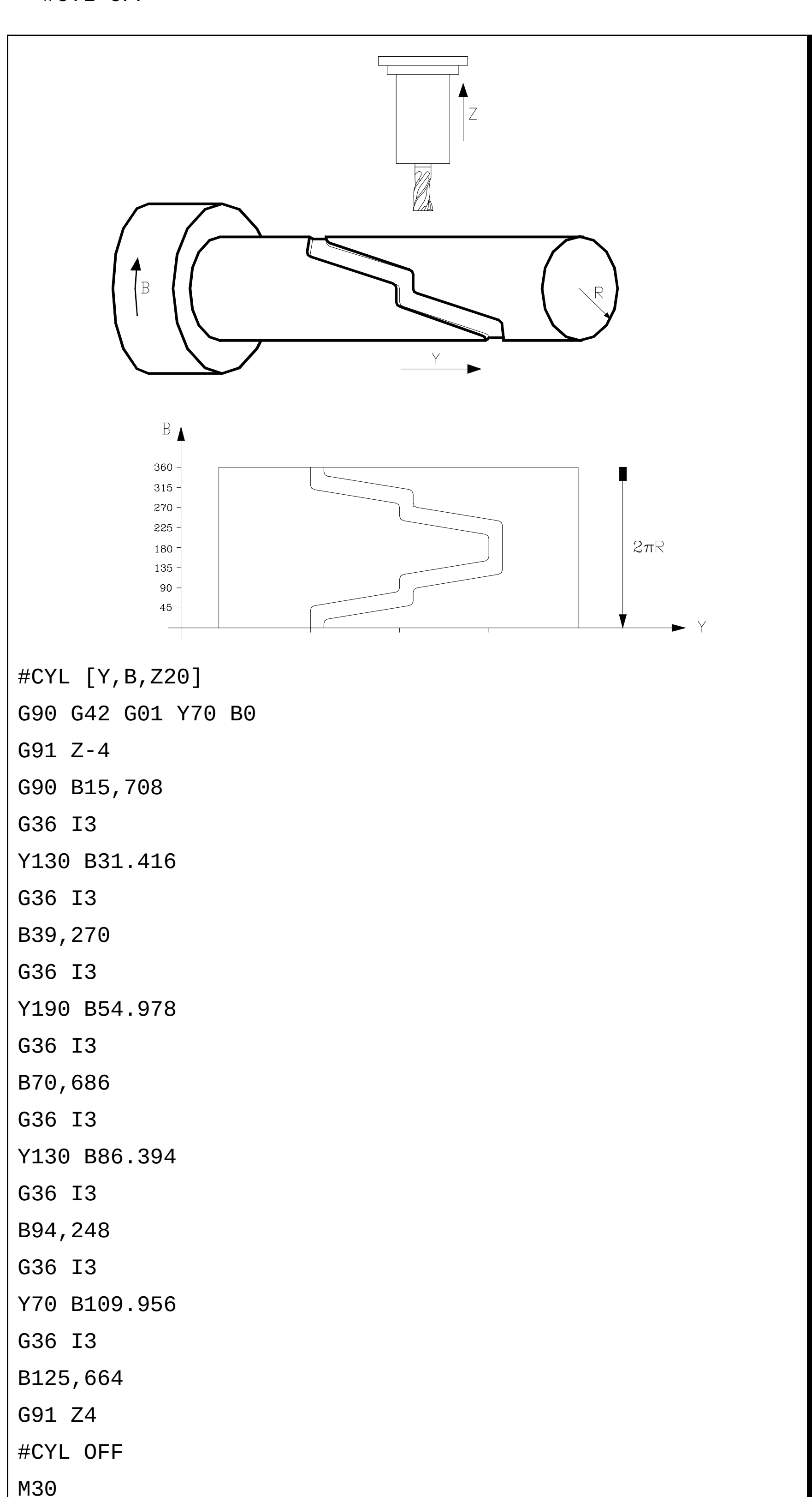


#CYL OFF 取消零件回转面加工

取消零件回转面加工。

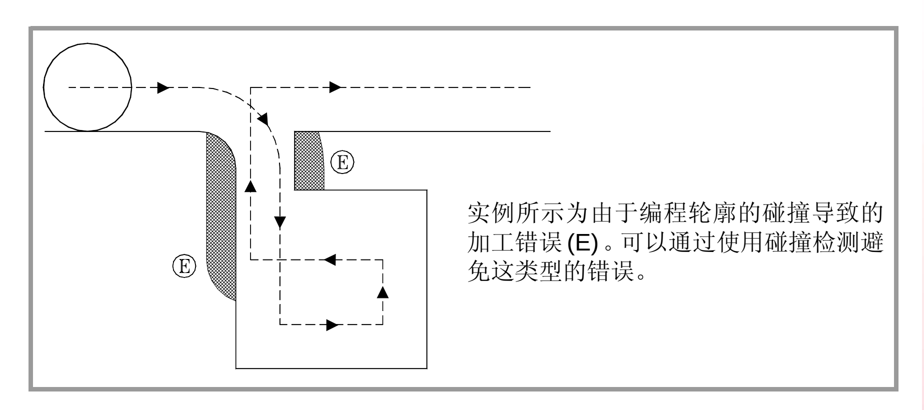
编程格式：

```
#CYL OFF
```



16.1.20 碰撞检测

用这个选项，数控系统预先分析要执行的程序段，以检测回路（轮廓之间的交叉点）或编程轮廓的碰撞。操作者可以定义最多 200 个将待分析的程序段。



实例所示为由于编程轮廓的碰撞导致的加工错误 (E)。可以通过使用碰撞检测避免这类型的错误。

当检测回路或碰撞时，数控系统不执行引起它的程序段，屏幕将显示报警，使操作者知道编程轮廓被修正。将对为每个回路或排除的碰撞显示警告。

包含在已删除的程序段的信息，如果不是有效平面内的运动，将被执行 (包括其它轴的运动)。

碰撞检测过程注意事项

- 碰撞检测可以应用，甚至当刀具半径补偿不起作用时。
- 碰撞检测有效，能够应用零点偏置，坐标预制和更换刀具。可是，不能应用原点搜索和测量。
- 更换工作平面将会打扰碰撞检测过程。数控系统在到目前为止存储的程序段检查碰撞，用开始新运动程序段的新平面恢复过程。
- 当编写包含同步程序段准备和执行 (例如 #FLUSH) 的指令（显式或隐式）时，碰撞检测过程将被中断。在执行那个指令之后恢复执行。
- 如果 Hirth 轴有效并且是主平面的组成部分，碰撞检测不能被激活。同样地，当碰撞检测有效时，轴不能作为 Hirth 轴被激活，并且如果工作平面的一根轴是 Hirth 轴，工作平面不能被更换。

#CD ON **激活碰撞检测**

激活碰撞检测过程。碰撞检测已经有效，修改要分析的程序段数量。

编程格式：

```
#CD ON [<blocks>]
```

参数	含义
<blocks>	可选参数。要分析的程序段数量。

定义要分析的程序段数量是可选项。如果没有定义，数控系统采用最大值 (200 程序段)。程序段范围可以在任何时候改变，甚至当碰撞检测有效时。

#CD OFF 取消碰撞检测

取消碰撞检测过程。

在执行 M02 或 M30 以及错误或复位之后，该过程将自动取消。

```
关于回路轮廓的实例
#CD ON [50]
G01 X0 Y0 Z0 F750
X100 Y0
Y -50
X90
Y20
X40
Y -50
X0
Y0
#CD OFF
```

```
轮廓碰撞实例
#CD ON
G01 G41 X0 Y0 Z0 F750
X50
Y -50
X100
Y -10
X60
Y0
X150
Y -100
X0
G40 X0 Y0
#CD OFF
M30
```

16.1.21 关于手动干涉

用该指令，当手动干涉有效时能够设置进给率和在 jog 方式的运动。用功能 G200, G201 和 G202 通过编程激活手动干涉。

用这些指令可以定义下列各项：

- 每种工作模式 （连续或增量 JOG）下的手动干涉轴进给率和手轮分辨率。

  这些值可以在激活手动干涉之前或之后定义，保持有效直到程序结束或复位。

- 附加手动干涉完成的运动极限。当指令编程运动时，这些极限被忽略。

  在激活手动干涉之后定义极限，保持有效直到极限被取消。

#CONTJOG 连续 JOG

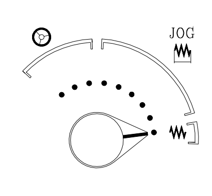
为连续 JOG 指令定义指定轴进给率。

编程格式：

```
#CONTJOG [<F>] <Xn>
```

参数	含义
<F>	进给率。
<Xn>	轴。

根据有效单位，进给率以单位毫米 / 分钟或英寸 / 分钟编程。




```
...
N100 #CONTJOG [400] X      连续 JOG， X 轴进给率。
N110 #CONTJOG [600] Y      连续 JOG， Y 轴进给率。
N120 G201 #AXIS [X,Y]
...
```

#INCJOG 增量 JOG

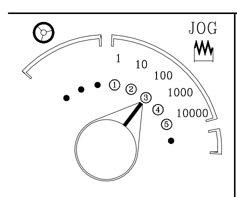
该指令定义指定增量运动和每个选择开关的增量 JOG 位置的轴进给率。

编程格式：

```
#INCJOG [<inc1>,<F>]...[<inc10000>,<F>] <Xn>
```

参数	含义
<inc>	每个增量 jog 的位置增量
<F>	每个增量 jog 的位置进给率。
<Xn>	轴

根据有效单位制，进给率以单位毫米 / 分钟或英寸 / 分钟编程。运动以毫米或英寸编程。




```
...
N100 #INCJOG [[0.1,100][0.5,200][1,300][5,400][10,500]] X
N110 G201 #AXIS [X]
...
```

在每个位置 X 轴的进给率和运动是 :

(1) 0.1mm a 100mm/min.
(2) 0.5mm a 200mm/min.
(3) 1mm a 300mm/min.
(4) 5mm a 400mm/min.
(5) 10mm a 500mm/min.

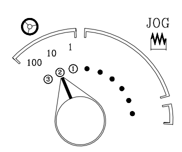
#MPGRESOL 手轮

该指令定义给定轴，在选择开关的每个位置，每一手轮脉冲的距离。

编程格式：

```
#MPGRESOL [<pos1>,<pos2>,<pos3>] <Xn>
```

参数	含义
<pos>	每个手轮位置的分辨率。
<Xn>	轴。




```
...
N100 #MPGRESOL [0.1,1,10] X
N110 G201 #AXIS [X]
N120 #MPGRESOL [0.5] Y
...
```

在每个位置每一 X 轴手轮脉冲的距离 :

(1) 手轮的 0.1 毫米 / 转。

(2) 手轮的 1 毫米 / 转。

(3) 手轮的 10 毫米 / 转。



该指令设置每个手轮脉冲的距离，在一个等于数控系统循环时间的时间段内。如果运动所需的进给率超过机床厂商设置的最大值，进给率将被限制在该值，并且轴运动距离将小于指令所编写的。

例如 : 如果编写 5 mm 的运动并且循环时间是 4 毫秒，结果速度是 1250 毫米 / 秒。如果最大进给率极限是 1000 毫米 / 秒，实际运动距离将是 4 mm。

#SET OFFSET 极限

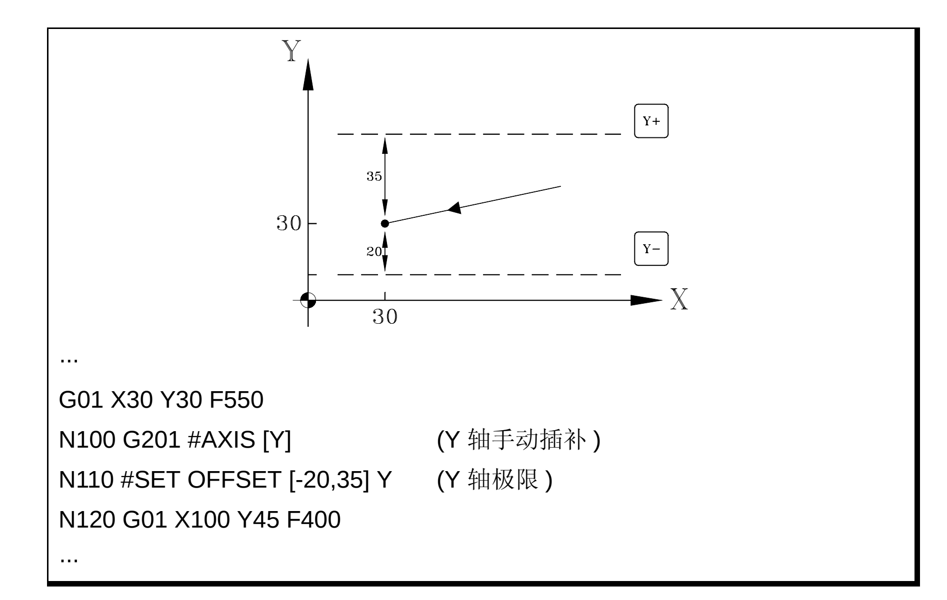
该指令定义指定轴的上限和下限，在上下限之内，在附加手动干涉期间轴可以微动。

编程格式：

```
#SET OFFSET [<lower>, <upper>] <Xn>
```

参数	含义
<lower>	下限
<upper>	上限
<axis>	轴

极限都是相对于轴位置的。下极限必须小于或等于零点，上极限必须大于零点 。




```
...
G01 X30 Y30 F550
N100 G201 #AXIS [Y]            (Y 轴手动插补 )
N110 #SET OFFSET [-20,35] Y    (Y 轴极限 )
N120 G01 X100 Y45 F400
...
```

#SYNC POS 同步

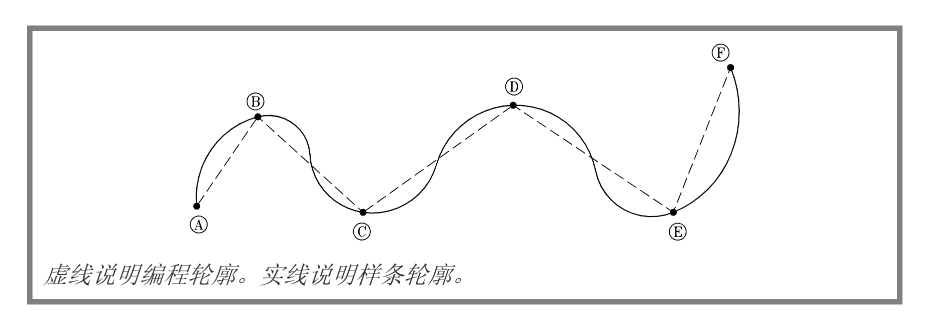
该指令使准备坐标和执行坐标同步，并且采用附加手动偏置。

编程格式：

```
#SYNC POS
```

16.1.22 样条 (Akima)

该类型的加工修改编程轮廓为样条类型的曲线，曲线通过所有编程点。



虚线说明编程轮廓。实线说明样条轮廓。

用直线(G00/G01)路径定义将要修改为样条曲线的轮廓。当定义圆弧(G02/G03)时，加工它并且它在下一直线路径恢复时，样条被中断。圆弧和样条之间用切线过渡。

#SPLINE ON **激活样条修正**

当执行该指令时，数控系统认为下一个编程点是样条的一部分并且开始产生曲线。

编程格式：

```
#SPLINE ON
```

如果刀具半径补偿在程序段之间线性过渡 (G136)，样条加工不能激活。反之亦然。

#SPLINE OFF **取消样条修正**

当执行该指令时，数控系统结束样条，以编程路径继续加工。

编程格式：

```
#SPLINE OFF
```

至少编写三个点，样条才能取消。当定义样条的初始和结束切线时，两个点足够了。

#ASPLINE MODE **选择切线类型**

该指令设置样条的初始和结束切线的类型，切线类型决定从上一路径到下一路径的过渡。它是可选参数；如果没有定义，切线被自动计算。

编程格式：

```
#ASPLINE MODE [<initial>,<final>]
```

参数	含义
<initial>	初始切线。
<final>	结束切线。

样条的初始和结束切线可以取下列值中的一个。如果没有编写，取值 1 。

值	含义
1	自动计算切线。
2	和上一 / 下一程序段相切的切线。
3	指定切线。

如果用值·3·定义，用指令#ASPLINE STARTTANG定义初始切线，用指令#ASPLINE ENDTANG 定义结束切线。如果没有定义，应用上一次使用的值。

#ASPLINE STARTTANG

初始切线

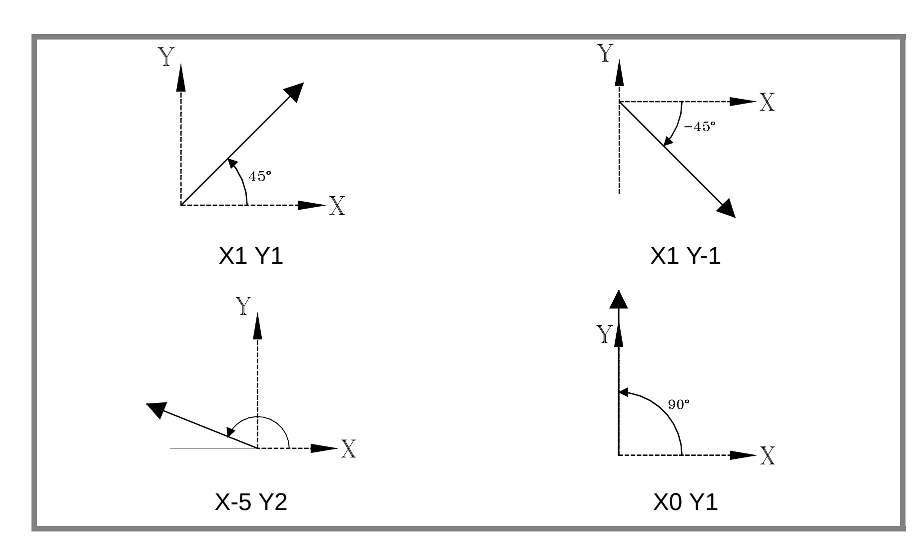
#ASPLINE ENDTANG

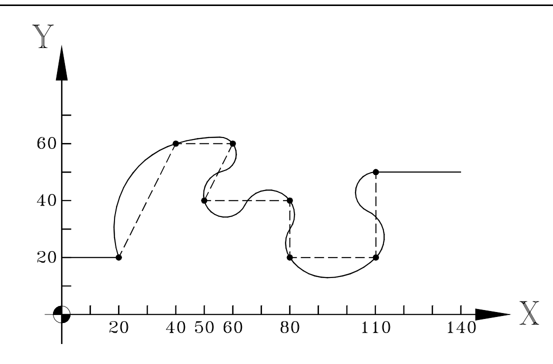
结束切线

这些指令定义样条的初始和结束切线。切线由给出的沿不同轴的矢量方向决定。

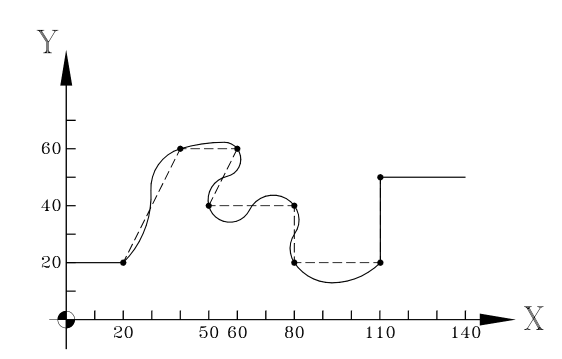
编程格式：

```
#ASPLINE STARTTANG <axes>
#ASPLINE ENDTANG <axes>
```






```
N10 G00 X0 Y20
N20 G01 X20 Y20 F750                  (样条的起始点)
N30 #ASPLINE MODE [1,2]               (初始和结束切线的类型)
N40 #SPLINE ON                        (激活样条)
N50 X40 Y60
N60 X60
N70 X50 Y40
N80 X80
N90 Y20
N100 X110
N110 Y50                              (样条的结束点)
N120 #SPLINE OFF                      (取消样条)
N130 X140
N140 M30
```




```
N10 G00 X0 Y20
N20 G01 X20 Y20 F750                  (样条的起始点)
N30 #ASPLINE MODE [3.3]               (初始和结束切线的类型)
N31 #ASPLINE STARTTANG X1 Y1
N32 #ASPLINE ENDTANG X0 Y1
N40 #SPLINE ON                        (激活样条)
. . .
N120 #SPLINE OFF                      (取消样条)
N130 X140
N140 M30
```

16.1.23 多项式插补

数控系统允许插补直线和圆弧，指令 #POLY 可以用于插补复杂的曲线，例如抛物线。

#POLY

多项式插补

该种类型的插补允许加工用多项式给出的曲线，多项式允许最高的次数是 4 次，插补参数是圆弧的长度。

编程格式：

```
#POLY [<eje>[a,b,c,d,e]...SP<sp> EP<ep>
```

参数	含义
<axis>	插补轴。
a,b,c,d,e	多项式系数。
<sp>	插补的起始参数。
<ep>	插补的结束参数。

必须定义所有要插补的轴和依次对应系数。

$$a + b\cdot<轴> + c\cdot<轴>^2 + d\cdot<轴>^3 + e\cdot<轴>^4$$

编写抛物线。多项式可以如下表示：

X 轴系数 : [0,60,0,0,0]

Y 轴系数 : [1,0,3,0,0]

起始参数 : 0

结束参数 : 60

```
G0 X0 Y0 Z1 F1000
G1
#POLY [X[0,60,0,0,0] Y[1,0,3,0,0] SP0 EP60]
M30
```

16.1.24 高速加工

现在，大多数零件使用 CAD-CAM 系统来设计。这种类型的信息是生成 CNC 程序的后置处理，通常由大量的几毫米或仅几十分之一微米的非常短的程序段组成。

随于此类零件加工，数控系统必须能够预先分析大量的点，这样才能生成连续路径，当保持 （可能的最佳路径）编程进给率和每根轴与路径的最大加速度、加加速度等等的限制时，路经通过 （或接近）编程点。

要执行由多个小程序段组成的程序的命令，典型的就是高速加工，用单独的指令 `#HSC ON` 来执行。该指令的参数是允许的最大轮廓误差。从该指令开始，数控系统修改几何尺寸，通过智能算法除去不需要的点，自动生成样条和程序段间的多项式过渡。

弦误差设置

如早先提到的，CNC 引起的编程零件和生成零件之间的误差决不大于编程值。另一方面，当加工初始零件并且转换路径成为 CNC 程序时， CAM 系统同样产生误差。最终误差可能是两者的和；因此，想得到的最大误差必定扩展到两个过程之间。

当生成程序时选择大的弦误差，当执行时选择小的弦误差，执行越慢质量越低。在这种情况脊会出现，因为数控系统完全地跟随 CAM 生成的多面体。推荐后置处理使用小于用于高速切削 HSC 的误差 (10% 到 20% 之间)。例如，对于 50 微米的最大误差，可以用 5 或 10 的误差，用 HSC 编写其余的。对于 CNC，较大的余量允许修改轮廓，从而可考虑每根轴的动力学影响，以不会导致不想要的结果，例如脊。

最后，自从 CNC 以纳米的精确度工作以来，可以获得更好的结果，坐标带 4 或 5 位小数的结果肯定要优于带 2 或 3 位小数的结果。这个没有负面影响，因为程序段加工时间不会发生显著地改变。程序尺寸的微小增加也不是什么问题，因为现在可使用高性能的硬盘来存储，或用以太网直接传送它们。

#HCS ON 激活高速加工

激活 HSC 模式，该模式允许执行由大量小程序段组成的程序，典型的高速加工。

编程格式：

```
#HCS ON [CONTERROR <error>]
```

参数	含义
<error>	可选参数。允许的最大轮廓误差。

该指令的参数是编程路径和结果路径之间允许的最大轮廓误差。该参数是可选参数，如果没有编写，采用机床参数 `MAXROUND` 设置的值作为最大轮廓误差。

#HCS OFF **取消高速加工 (切削)**

取消高速加工模式。

编程格式：

```
#HCS OFF
```

当编写功能 G05, G07 或 G50 中的任何一个时，也可以取消。

16.1.25 加速度控制

用机床参数设置应用在运动中的加速度和加加速度。然而，也可以用功能 G130, G131, G132 和 G133 从程序改变这些值。

指令 #SLOPE 设置由这些功能定义的值的影响，即加速度是如何运行的。

#SLOPE

设置加速度运行方式

该指令设置用功能 G130-G133 定义的值对加速度运行方式的影响。

编程格式：

```
#SLOPE [<type>,<jerk>,<accel>,<move>]
```

参数	含义
<type>	加速度类型。
<jerk>	可选参数。设置加加速度影响。
<accel>	可选参数。设置加速度影响。
<mov>	可选参数。影响 G00 的运动。

```
#SLOPE [1,1,0,0]
#SLOPE [1]
#SLOPE [2,,,1]
```

不需要编写所有的参数。每个参数允许的取值如下。

- 参数 <type> 决定加速度的类型 。

值	含义
0	线性加速度。
1	梯形加速度。
2	方正弦形 (钟形) 加速度。

缺省，采取 0 值。

可选参数 <jerk> 设置用功能 G132 和 G133 定义的加加速度的影响。仅在梯形和方正弦形加速度时考虑该参数。

值	含义
0	修改加速阶段和减速阶段的加加速度和减加速度。
1	修改加速阶段的加加速度。
2	修改减速阶段的减加速度。

缺省，采用 0 值。

- 可选参数 <accel> 设置用功能 G130 和 G131 定义的加速度的影响 。

值	含义
0	一直应用。
1	仅应用在加速度阶段。
2	仅应用在减速度阶段。

缺省，采用 0 值。

- 可选参数 I <move> 决定功能 G130, G131, G132 和 G133 是否影响 G00 运动。

值	含义
0	影响 G00 运动。
1	不影响 G00 运动。

缺省，采用 0 值。

16.1.26 坐标变换

这部分说明和坐标变换相关的指令。在本手册的 “13 坐标变换”章详细说明这些指令的编程方法和工作方式。

#KIN ID

运动学选择

用于选择主轴的运动学，定义使用的主轴类型，特性和尺寸。

编程格式：

```
#KIN ID [<num>]
```

参数	含义
<num>	可选参数。要激活的运动学编号。

如果没有编写，数控系统采用机床参数设置的缺省运动学。

#CS

定义和选择在倾斜平面的机床坐标系

#ACS

定义和选择在倾斜平面的夹具坐标系

用指令 #CS，可以定义、存储、激活和取消最多 5 个坐标系。用指令 #ACS ，可以定义、存储、激活和取消最多 5 个夹具坐标系。

两个指令用同样的编程格式，可以一起或分开使用。和指令相关参数有下列含义。

参数	含义
[n]	坐标系编号 (1..5)。
MODE m	应用的定义方式 (1..6)。
V1...V3	变换向量的组成。
φ1...φ3	旋转角度。
0/1	在模式 3, 4 和 5 排列轴。

编程格式：

- 定义和存储新的 #CS 或 #ACS。

```
#CS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- 定义、存储和激活新的 #CS 和 #ACS。

```
#CS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- 定义和激活 (不存储) 新的 #CS 或 #ACS。仅其中的一个可以被定义；定义另一个，前一个必须被取消。

```
#CS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- 取消和删除所有当前 #CS 或 #ACS，定义、存储和激活新的指令。

```
#CS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- 取消和删除所有当前 #CS 或 #ACS，定义、和激活（不存储）新的指令。

  #CS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

  #ACS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
- 采取和存储当前坐标系作为新的 #CS 或 #ACS。

  #CS DEF ACT [n]

  #ACS DEF ACT [n]
- 激活最近存储的 #CS 或 #ACS 。

  #CS ON

  #ACS ON
- 激活存储的 #CS 或 #ACS。

  #CS ON [n]

  #ACS ON [n]
- 取消最近激活的 #CS 或 #ACS 。

  #CS OFF

  #ACS OFF
- 取消所有激活的 #CS 或 #ACS。

  #CS OFF ALL

  #ACS OFF ALL

#RTCP ON **激活 RTCP (旋转刀具中心点) 变换**

#RTCP OFF **取消 RTCP (旋转刀具中心点) 变换**

用 RTCP 变换，刀具可以不需要改变相对零件的刀尖位置进行定位。

编程格式：

```
#RTCP ON
#RTCP OFF
```

当 TLC 有效时，不能选择功能 RTCP。

#TOOL ORI **刀具垂直于工作平面**

放置刀具垂直于工作平面。在第一个运动程序段之后发生。

编程格式：

```
#TOOL ORI
```

#TLC ON **激活刀具长度补偿**

#TLC OFF **取消刀具长度补偿**

CAD-CAM 程序考虑刀具长度补偿，生成刀具基准坐标。

使用不同尺寸的刀具加工时，功能 #TLC 补偿实际刀具长度和理论刀具长度之间的差值。

编程格式：

```
#TLC ON [n]
#TLC OFF
```

参数	含义
[n]	刀具长度差值 (实际 - 理论)

当功能 RTCP 有效时，不能选择功能 TLC 。

16.1.27 宏定义

宏可以在格式 "MacroName" = "CNCblock" 用它们自己的名称定义程序段或部分程序段。一旦定义了宏，编写 "MacroName" 与编写 "CNCblock" 相同。当从程序（或 MDI）执行宏时，数控系统执行相关程序段。

通过程序 （或 MDI）定义的宏储存在 CNC 表格里； 这种情况下，它们对程序的其余部分是可用的，不需要再定义它们。数控系统上电时初始化表格，也可以从局部程序用指令 #INIT MACROTAB 初始化，从而删除保存的宏。

#DEF: 宏定义

数控系统最多可以定义 50 个不同的宏 。可以从任何程序定义宏。当设法定义更多的宏时，数控系统发送相关错误信息。可以用指令 #INIT MACROTAB 初始化宏表格 （除去所有宏）。

宏定义必须单段编写。

编程格式：

```
#DEF "MacroName" = "BloqueCNC"
```

参数	含义
MacroName	在程序中用于识别宏的名称。由字母或数字组成，可以最多用 30 个字符。
CNCBlock	程序段。最长可以有 140 个字符。

如下在一个程序段可以定义多个宏。

```
#DEF "Macro1"="Block1" "Macro2"="Block2" ...
```

```
(宏定义)
#DEF "READY"="G0 X0 Y0 Z10"
#DEF "START"="SP1 M3 M41" "STOP"="M05"
(宏执行)
"READY" (同编写 G0 X0 Y0 Z10)
P1=800 "START" F450 (同编写 S800 M3 M41)
G01 Z0
X40 Y40
"STOP" (同编写 M05)
```

宏的算术操作定义

当在宏的定义里包含算术操作时，必须包括完整的算术操作。

```
宏的正确定义例子。
    #DEF "MACRO1"="P1*3"
    #DEF "MACRO2"="SIN [\"MACRO1\"]"
```

```
宏的错误定义例子。
    #DEF "MACRO1"="56+"
    #DEF "MACRO2"="12"
    #DEF "MACRO3="\"MACRO1\"\"MACRO2\""

    #DEF "MACRO4"="SIN["
    #DEF "MACRO5"="45]"
    #DEF "MACRO6="\"MACRO4\"\"MACRO5\""
```

宏的连接。包含在其它宏定义的宏。

宏的定义可以包含其它宏。在这种情况下，包含在定义里的宏的一个必须用符号 "\" 标界线 (\"macro\")。

```
例 1
#DEF "MACRO1"="X20 Y35"
#DEF "MACRO2"="S1000 M03"
#DEF "MACRO3"="G01 \"MA1\" F100 \"MA2\""
例 2
#DEF "POS"="G1 X0 Y0 Z0"
#DEF "START"="S750 F450 M03"
#DEF "MACRO"="\"POS\" \"START\""
```

#INIT MACROTAB重置宏表格

当从程序 (或 MDI) 定义宏时，它被存储在 CNC 表中，所以它对程序其余部分是可用的。该指令重置宏表格，擦除存储在里面的宏。

16.1.28 程序段重复

该指令可以用于执行定义在两程序段之间的部分程序，两程序段可以用标签识别。后一程序段的标签必须单段编写。

可根据需要，定义重复执行次数；如果没有定义，将重复一次。

从指令被执行的相同的程序或子程序，必须定义要重复的程序段号。它们也可以位于程序之后 (功能 M30 之后)。

最多允许 20 级嵌套。

#RPT

程序段重复

编程格式：

```
#RPT [<blk1>,<blk2>,<n>]
```

参数	含义
<blk1>	第一程序段。
<blk2>	末尾程序段。
<n>	可选参数。重复次数。

因为识别程序段的标签可以是两种类型（编号和名称），指令 #RPT 可以如下编程 :

- 标签是程序段号。

  在包含首和尾标签的程序段，在程序段号后编写符号 ":" 。在每个跳转目标的标签处是必需的。

```
N10 #RPT [N50,N70]

N50: G01 G91 X15 F800          ( 第一程序段 )
X-10 Y-10
X20
X-10 Y10
N70:                           ( 末尾程序段 )
```

- 标签是程序段名称。

```
N10 #RPT [[BEGIN],[END]]

[BEGIN] G01 G91 F800           ( 第一程序段 )
X-10 Y-10
X20
X-10 Y10
G90
[END]                          ( 末尾程序段 )
```

一旦执行重复，执行从包含指令 #RPT 的程序段之后的程序段重新开始。

注意事项

首末程序段的标签必须是不同的。一个单独程序段的重复执行，编程如下：

```
N10 #RPT [N10,N20,4]

N10: G01 G91 F800              ( 第一程序段 )
N20:                           ( 末尾程序段 )
```

也能够用命令 "NR" 完成程序段重复执行。参见第 5 也的 “ISO 代码编程”。

如果控制回路的开始不在重复指令内，不能够重复一组结束控制回路的程序段。

```
N10 #RPT [N10,N20]

N10: $FOR P1=1,10,1
G0 XP1
$ENDFOR
G01 G91 F800
N20:
```



```
%PROGRAM
G00 X-25 Y-5
N10: G91 G01 F800          ( 定义轮廓 "a")
X10
Y10
X -10
Y -10
G90
N20:
G00 X15
#RPT [N10, N20]            ( 程序段重复。轮廓 "b")
#RPT [[INIT], [END], 2]    ( 程序段重复。轮廓 "c" 和 "d")
M30

[INIT]
G1 G90 X0 Y10
G1 G91 X10 Y10
X -20
X10 Y-10
G73 Q180
[END]
```

16.1.29 通道间的通信和同步

每个通道可以执行它自己的程序，和其它通道同时地或独立地执行。但是，除此之外，也能和其它通道通信，传递信息或在特殊点同步。

通信发生在每个通道的局部程序控制的标志的编号的基础 T 之上。这些标志确定通道是等待同步还是可以同步，等等。

同步有两种不同的方法，每种提供不同的解决方法。

* 用指令 #MEET。

  同步最容易的方法。在同步包括的所有通道停止执行。

  在执行 M02 或 M30 以及复位或上电后，初始化正在使用的标志集。

* 用指令 #WAIT - #SIGNAL - #CLEAR。

  这种方法比前一种更复杂，但是功能更多。为了同步不停止在所有通道的执行。

  在执行 M02 或 M30 以及复位或上电后，保持正在使用的标志集。

两种方法的同步标志是彼此独立的。由指令 #MEET 控制的标志既不影响也不受其它的指令影响。

同步通道的其它方法

通用算数参数也能够用于通信和同步通道。通过从一个通道写入 某一值，稍后从另一通道读取它，能够设置在执行程序时要继续的条件。

从另一通道访问通道的变量也能够作为通信方式使用。

在通道间交换轴也能够同步，应为直到轴被另一个通道释放，通道才能攫取轴。

通道 1	通道 2	通道 3
G1 F1000	X1=0 Y1=0 Z1=0	G1 F1000
S3000 M3	G1 F1000	X2=20 Z2=10
#FREE AX [Z] (释放 Z 轴)	#FREE AX[Z1] (释放 Z1 轴)	#FREE AX[Z2] (释放 Z2 轴)
X30 Y0	G2 X1=-50 Y1=0 I-25	X2=100 Y2=50
#CALL AX [Z1,Z2] (增加 Z1 和 Z2 轴)	#CALL AX [Z] (增加 Z 轴)	#CALL AX[Z2] (恢复 Z2 轴)
X90 Y70 Z1=-30 Z2=-50	G1 X1=50 Z20	G0 X2=0 Y2=0 Z2=0
#FREE AX [Z1,Z2] (释放 Z1 和 Z2 轴)	#FREE AX[Z] (释放 Z 轴)	M30
X0	X1=20	
#CALL AX [Z] (恢复 Z 轴)	#CALL AX [Z1] (恢复 Z1 轴)	
G0 X0 Y0 Z0	G0 X1=0 Y1=0 Z1=0	
M30	M30	

查询

关于同步标志状态的信息可以用下列变量查询。

* 从 "m" 通道用 "n" 通道期待的 MEET 或 WAIT 型标志

  V.[n].G.MEETCH[m]

  V.[n].G.WAITCH[m]

  用通道编号替代字母 "n" 和 "m" 。

* 在 "n" 通道 MEET 或 WAIT 型 "m" 标志的状态

  V.[n].G.MEETST[m]

  V.[n].G.WAITST[m]

#MEET 激活在通道被指示的标志并且在编写的其它通道等待它激活

该指令，在自己的通道激活标志后，在恢复执行之前在被编程的通道等待它也有效。每个通道有 10 个标志，编号从 1 到 10。

在多个通道编写相同的指令，所有的都停止并等待其余的达到确定的点，它们都从那点同时恢复执行。

编程格式：

```
#MEET [<mark>, <channel>,...]
```

参数	含义
<mark>	在通道本身激活的同步标志以及继续执行之前在其余通道必须激活的同步标号。
<channel>	通道或相同标志必须被激活的通道。

不需要在每条指令中包括它本身通道的编号，因为当执行指令 #MEET 时标志被激活。但是为了程序具有更好的可读性，推荐编写它。

操作

在每个通道编写相同指令，所有的都在那点同步并且恢复执行。工作如下。

1. 在自己的通道激活所选的标志。
2. 在所有确定的通道等待要待激活的标志。
3. 在同步通道之后，从自己的通道删除标志并且继续执行程序。

每个通道在它的 #MEET 停止。当它们中的最后一个到达命令点并且检查所有标志有效后，处理过程同时向它们解锁。

在下述实例中，通道 ·1·、 ·2· 和 ·3· 等待要激活的标志 5，同步通道并恢复执行。

通道 1	通道 2	通道 3
%PRG_1	%PRG_2	%PRG_3
...	...	...
...	#MEET [5,1,2,3]	...
#MEET [5,1,2,3]	...	...
...	...	...
...	...	#MEET [5,1,2,3]
M30	M30	M30

#WAIT 在确定的通道等待将被激活的标志

在指定通道指令 #WAIT 等待要激活的指示标志。如果当执行该指令时标志已经是有效的，不会终止执行并且程序保持运转。

每个通道有 10 个标志，编号从 1 到 10。

编程格式：

```
#WAIT [<mark>, <channel>,...]
```

参数	含义
<mark>	等待激活的同步标志。
<channel>	通道或必须激活标志的通道。

和指令 #MEET 相反，不激活自己通道的指示标志。通道的标志用指令 #SIGNAL 激活。

#SIGNAL 在自己的通道激活标志

指令 #SIGNAL 在自己的通道激活指示标志 。每个通道有 10 个标志，编号从 1 到 10。这些标志 对应指令 #WAIT 。

该指令不执行任何等待，连续进行。一旦同步，取消标志，如果想要，用指令 #CLEAR。

编程格式：

```
#SIGNAL [<mark>,...]
```

参数	含义
<mark>	在通道被激活的同步标志。

#CLEAR 清除通道的同步标志

该指令在自己通道取消指示标志。如果没有编写标志，删除所有的标志。

编程格式：

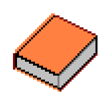
```
#CLEAR
#CLEAR [<mark>,...]
```

参数	含义
<mark>	在通道被删除的同步信号。

在下述实例中，通道 ·1· 和 ·2· 等待要激活的标志 5，在通道 ·3· 同步。当标志 ·5· 在通道 ·3· 被激活时，在所有三个通道恢复执行。

通道 1	通道 2	通道 3
%PRG_1	%PRG_2	%PRG_3
...	...	...
...	#WAIT [5,3]	...
#WAIT [5,3]	...	...
...	...	#SIGNAL [5]
...	...	...
...	...	#CLEAR [5]
M30	M30	M30

16.1.30 独立轴的运动



该功能有专用手册。

本部分手册仅提供一些和该功能有关的知识。想要获得更多有关独立轴的必要条件和操作的信息查询相关专用手册。

数控系统能够执行独立的定位和同步。对于这种类型的运动，每根数控轴有独立的插补器，保证当前点轨迹在它自己的位置，而不依赖数控系统的普通插补的轨迹。

能够同时执行独立的运动和常规运动。结果是两个插补的和。

数控系统存储最多一个每轴两个独立运动指令的最大值。当有两个未决的执行时，从局部程序提示等待，并发送其余的指令。

对独立轴的限制

用相关指令可以独立地移动通道的任何轴。然而，该功能有下列限制。

- 当在轴模式用指令 #CAX 设置时，主轴仅能独立运动。然而，它总是同步的主控主轴。
- 旋转轴可以是任何模式，但是最低速度极限必须总是零。
- Hirth 轴不能独立运动。

同步插补

为了对增量运动考虑机床的实际坐标，每个插补必须和实际坐标同步。用指令 #SYNC POS 从局部程序执行同步。

数控系统的复位使两种插补的理论坐标和实际坐标同步。这些同步仅在嵌入两种类型的插补的指令 时是必需的。

每次程序初始化或执行 MDI 方式的程序段，数控系统的一般插补的坐标同步，并且每个新独立指令 （没有任何一个是未决的）也和独立插补的坐标同步。

在程序段准备的运动的影响

这些程序段没有一个能中断程序段准备，但是它们中断插补。因此，不能混和两程序段，将有一个中间的程序段。

定位运动 (#MOVE)

各种类型的定位用下列指令编写。

```
#MOVE          - 绝对定位运动。
#MOVE ADD      - 增量定位运动。
#MOVE INF      - 无限定位运动。
```

每一种的编程格式如下。在符号 <> 之间为可选参数。

```
#MOVE <ABS> [Xpos <,Fn> <,blend>]
#MOVE ADD [Xpos <,Fn> <,blend>]
#MOVE INF [X+/- <,Fn> <,blend>]
```

[Xpos] 轴和要到达的位置

轴和要到达的位置。指令 #MOVE ABS 表示用绝对坐标定义。指令 #MOVE ADD 表示用增量坐标定义。

由坐标或编写的增量决定运动方向。对于旋转轴，由轴的类型决定运动方向。如果是标准的，通过最短路径；如果是单向的，以预制方向运动。

[X+/-] 轴和运动方向

要定位的轴 (无坐标) 。符号表示运动的方向。

与 #MOVE INF 一起使用，执行无限的运动，直到达到轴的极限或运动被中断。

[Fn] 定位进给率

定位进给率。

进给率以单位毫米 / 分钟、英寸 / 分钟或度 / 分钟给出。

可选参数。如果没有定义，采用机床参数 POSFEED 设置的进给率。

[blend] 与下一程序段动态混合

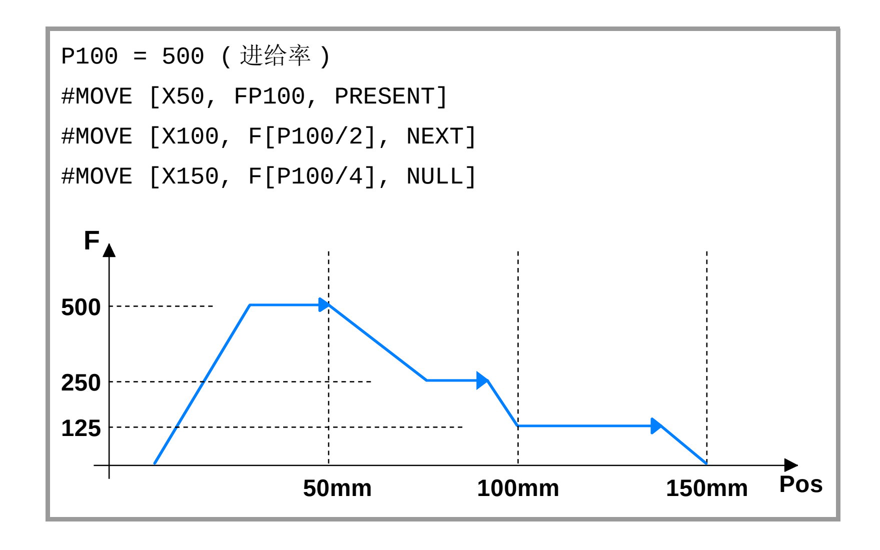
可选参数。用可选参数定义用于到达位置的进给率 （与下一程序段动态混合）。

用下列元素中的一个给出用于到达位置的进给率：

[混合]	动态混合的类型
PRESENT	以使用于程序段本身的定位进给率到达指定位置。
NEXT	以下一程序段指定的定位进给率到达指定位置。
NULL	以零进给率到达指定位置。
WAITINPOS	以零进给率到达指定位置，在执行下一程序段之前等待到到适当的位置。

编写该参数是可选的。如果没有编写，如下依照机床参数 ICORNER 执行动态混合。

ICORNER	动态混合的类型
G5	依据对 PRESENT 值的设置。
G50	依据对 NULL 值的设置。
G7	依据对 WAITINPOS 值的设置。



同步运动 (#FOLLOW ON)

用下列指令编写不同类型同步的激活和取消。

`#FOLLOW ON`	- 激活同步运动。
`#FOLLOW OFF`	- 取消同步运动。

每一种的编程格式如下。在符号 <> 之间表示可选参数。

```
#FOLLOW ON [master, slave, Nratio, Dratio <,synctype>]
#FOLLOW OFF [slave]
```

指令 `#FOLLOW OFF` 的执行包括消除从动轴的同步速度。轴将取得时间刹车，指令在此时间将保持指令。

[master] 主控轴

主要主轴的名称。

[slave] 从动轴

从动轴的名称。

[Nratio] 齿轮比 (从动轴)

齿数比的分子。从动轴的旋转。

[Dratio] Gear ratio (master axis)

齿数比的分母。主控轴的旋转。

[synctype] 同步类型

可选参数。确定是速度同步还是位置同步的标志。

[synctype]	同步类型
POS	位置同步。
VEL	速度同步。

该参数的编程是可选的。如果没有编写，执行速度同步。

```
#FOLLOW ON [X, Y, N1, D1]
#FOLLOW ON [A1, U, N2, D1, POS]
#FOLLOW OFF [Y]
```

16.1.31 附加编程指令

#COMMENT BEGIN注释开始

#COMMENT END 注释结束

指令 #COMMENT BEGIN 和 #COMMENT END 指示注释的开始和结束 。

编程格式：

```
#COMMENT BEGIN
#COMMENT END
```

数控系统认为在它们之间编写的程序段是单独的注释，当执行程序时忽略。

```
#COMMENT BEGIN
   P1 : 加工宽度
   P2 : 加工长度
   P3 : 加工深度
#COMMENT END
```

#FLUSH 程序段准备中断

数控系统在执行一个程序段之前读取多个程序段，为了预先计算跟随的路径。

指令 #FLUSH 预先中断程序段准备，执行上一准备程序段，同步程序段的准备和执行，继续程序。当恢复时，再次预先开始准备 。

编程格式：

```
#FLUSH
```

读取它时程序段有待分析的数据；执行时分析它，然后使用指令 #FLUSH 。

该指令用于在执行的时候计算 " 程序段跳转 " 条件。

```
...
N110 #FLUSH
/N120 G01 X100
...
```

必须记住中断程序段准备可以导致补偿路径不同于编程路径，当工作在非常短的运动、加加速轴运动等等时，出现不想要的转接。

#WAIT FOR 等待结果

该指令中断程序执行直到条件合适。

编程格式：

```
#WAIT FOR [<condition>]
```

```
#WAIT FOR [V.PLC.O[1] == 1]
```

能比较数字、参数或结果是数字的算数表达式。

#SELECT PROBE 探针选择

可以用于选择探针。

编程格式：

```
#SELECT PROBE [<probe>]
```

值	含义
1	探针 1
2	探针 2

当有多个探针安装在机床上时才需要该指令。

#TANGFEED RMIN恒切线进给率

当应用恒切线进给率 (G196) 时，用该指令可以设置最小半径，所以该类型的进给率进应用在半径大于最小值的弧。

编程格式：

```
#TANGFEED RMIN [<radius>]
```

如果不编写或设置为零，数控系统对所有弧应用恒切线进给率。

本手册的 “5 工艺功能” 章节进一步详细说明如何操作恒切线进给率。

#ROUNDPAR 圆角

使用该指令，能够选择和定义要应用的圆角类型。有 5 种类型的圆角。

该指令可以最多有 6 个和它相关的参数，它们的含义跟选择的拐角类型有关。

本手册中的 “7 几何协助” 章节进一步详细说明如何定义圆角。

#TIME 定时

为指示时间周期 （单位秒）中断程序的执行。

编程格式：

```
#TIME [<time>]
```

当时间用常量或参数编写时括弧可以省略。

```
P1=20
#TIME [P1+2]
    (22 秒暂停 )
#TIME 5
    (5 秒暂停 )
```

暂停也可以用功能 G04 编程，在手册的 “8 附加准备功能” 章详细介绍。

#SCALE 缩放比例因子

可用于放大或缩小编程的零件。这种方法，能够只用一个程序加工有相同外形但不同尺寸的成批零件。作用同功能 G72。

编程格式：

```
#SCALE [<scale>]
```

激活缩放比例因子之后，所有编程坐标被乘以定义的比例系数，直到定义新的比例因子或被取消 (编写缩放比例因子 "1")。

本手册的 “7 几何协助” 章节进一步详细说明如何定义缩放比例因子。

16.2 流程控制指令

16.2.1 程序段跳转 ($GOTO)

$GOTO N< 表达式 >
$GOTO [< 标志 >]

在该指令中，用下列参数中的一个进行定义：

< 表达式 >	可以是数字、参数或结果是数字的算数表达式。
< 标志 >	可以是最多 14 个大写字母、小写字母或数字的字符组成的序列 (不允许有空格或引号)。

该指令激活程序段跳转到用 "N< 表达式 >" 或 "[< 标志 >]" 定义的程序段，程序段可以是定义在指令 $GOTO 之前或之后的位置。在跳转到指定的程序段之后继续程序执行。

指令 $GOTO 可以在两种方式编程：

• 用程序段号。

在这些跳转的目标程序段 ，编写标志必须后加 ":"。

目标	N< 数字 >:
调用	$GOTO N< 数字 > or N< 数字 >:

• 用标志。

目标	[< 标志 >]
调用	$GOTO [< 标志 >]

调用指令和目标程序段必须在同一程序或子程序内。

N10 $GOTO N60	N40:	N10 $GOTO [LABEL]
...	...	...
N60: ...	N90 $GOTO N40:	N40 [LABEL]

不能有跳转程序段包含在另一指令中 ($IF, $FOR, $WHILE, 等等) 。

尽管流程控制指令必须在程序段单独编写，在同一程序段里指令 $GOTO 可以增加指令。这种方式，能够不用结束回路，退出包含指令 ($IF, $FOR, $WHILE, 等等) 的程序段。

```
N10 P0=10
N20 $WHILE P0<=10
N30  G01 X[P0*10] F400
N40  P0=P0-1
N50  $IF P0==1 $GOTO N100
N60  $ENDWHILE
N100: G00 Y30
M30
```

16.2.2 条件执行 ($IF)

$IF < 条件 >... $ENDIF

该指令中定义下列参数：

< 条件 > 可以比较两个数字、参数或结果是数字的算数表达式。

该指令分析编写的条件。

- 如果条件为真，执行包含在指令 $IF 和 $ENDIF 之间的程序段。
- 如果条件不为真，执行包含在指令 $ENDIF 之后的程序段。

```
...
N20 $IF P1==1
 N30...
 N40...
N50 $ENDIF
N60 ...
```

如果 P1 等于 1, 将执行程序段从 N30 到 N40。

如果 P1 不等于 1, 执行从 N60 继续。

指令 $IF 总是用 $ENDIF 结束，除了当增加指令 $GOTO，在这种情况下，不必编写。

```
...
N20 $IF P1==1 $GOTO N40
N30...
N40: ...
N50...
```

如果 P1 等于 1, 执行从 N40 继续。

如果 P1 不等于 1, 执行从 N30 继续。

作为可选项，指令 $ELSE 和 $ELSEIF 可以插在 $IF 和 $ENDIF 之间。

$IF < 条件 > ... $ELSE ... $ENDIF

该指令分析编写的条件。

- 如果条件为真，执行包含在 $IF 和 $ELSE 之间的程序段，并执行在 $ENDIF 之后的程序段继续。
- 如果条件不为真，执行包含在 $ELSE 和 $ENDIF 之间的程序段。

```
N20 $IF P1==1
 N30...
 N40...
N50 $ELSE
 N60...
 N70...
N80 $ENDIF
N90 ...
```

如果 P1 等于 1, 将执行程序段从 N30 到 N40。执行在 N90 继续。

如果 P1 不等于 1, 执行在 N50 继续。

$IF < 条件 1>... $ELSEIF< 条件 2>... $ENDIF

该指令分析下列编写的条件。

- 如果 < 条件 1> 是真，执行包含在 $IF 和 $ELSEIF 之间的程序段 。
- 如果<条件1>不为真,分析<条件2>。如果为真，执行包含在$ELSEIF 和$ENDIF 之间的程序段 (或即便要 $ELSEIF)。
- 如果所有条件不为真，执行从 $ENDIF 之后的程序段继续。

指令 $ELSEIF 的多少根据需要的情况确定。

```
N20 $IF P1==1
N30...
N40...
N50 $ELSEIF P2==[-5]
N60...
N70 $ELSE
N80...
N90 $ENDIF
N100 ...
```

- *如果 P1 等于 1, 将执行程序段从 N30 到 N40。执行在 N100 继续。*
- *如果 P1 不等于 1 并且 P2 等于 -5，执行程序段 N60。执行在 N100 继续。*
- *如果P1 不等于1 并且 P2 不等于 -5，执行程序段N80 并且执行在 N100 继续。*

执行指令 $ELSE 也可以包含在里面。在这种情况下，如果所有条件不为真，执行在 $ELSE 和 $ENDIF 之间的程序段。

16.2.3 条件执行 ($SWITCH)

$SWITCH < 表达式 n1>... $CASE< 表达式 2>... $ENDSWITCH

该指令中定义下列参数：

< 表达式 > 可以是数字、参数或结果是数字的算数表达式。

该计算 < 表达式 1> 的结果并执行包含在和指令 $CASE 之间的程序段，它的 < 表达式 2> 有相同的值作为计算结果并且执行相应的指令 $BREAK。

指令 $SWITCH 总是以指令 $ENDSWITCH 结束。

指令 $CASE 总是以指令 $BREAK 结束 。可以按照需要编写多少个指令 $CASE 。

作为可选项，指令 $DEFAULT 可以以这样的一种方式插入，当 < 表达式 1> 的结果不和任何 < 表达式 2> 相一致时，执行包含在指令 $DEFAULT 和 $ENDSWITCH 之间的程序段。

```
N20 $SWITCH [P1+P2/P4]
N30 $CASE 10
N40...
N50...
N60 $BREAK
N70 $CASE [P5+P6]
N80...
N90...
N100 $BREAK
N110 $DEFAULT
N120...
N130...
N140 $ENDSWITCH
N150...
```

如果表达式 [P1+P2/P4] 的结果。

- *是 "10"，执行程序段 N40 到 N50。执行在 N150 继续。*
- *等于 [P5+P6], 执行程序段 N80 到 N90。执行在 N150 继续。*
- *不等于 "10" 和 [P5+P6], 执行程序段 N120 到 N130.。执行在 N150 继续。*

16.2.4 程序段重复 ($FOR)

$FOR <n> = < 表达式 1>,< 表达式 2>,< 表达式 3>... $ENDFOR

该指令中定义下列参数。

<n>	可以是可写变量的算数参数。
< 表达式 >	可以是数字、参数或结果是数字的算数表达式。

当执行该指令时，<n> 接受 < 表达式 1> 的值，改变它的值等于 < 表达式 2> 的值，同样次序地重复<表达式3>确定的次数。在每一步，执行包含在 $FOR 和$ENDFOR 之间的程序段。

```
...
N20 $FOR P1=0,10,2
N30...
N40...
N50...
N60 $ENDFOR
N70...
```

执行程序段 N30 到 N50 ，从 P1=0 到 P1=10, 2 次同样的次序 。(因而 6 次)

```
...
N12 $FOR V.P.VAR_NAME=20,15,-1
N22...
N32...
N42 $ENDFOR
N52...
```

执行程序段 N22 到 N32，以 V.P.VAR_NAME=20 到 V.P.VAR_NAME=15 的次序。(因而 5 次)

指令 $BREAK 让结尾程序段重复，当停止条件不满足时。程序的执行在程序段 $ENDFOR 之后继续 。

```
...
N20 $FOR P1= 1,10,1
N30...
N40 $IF P2==2
N50 $BREAK
N60 $ENDIF
N70...
N80 $ENDFOR
...
```

如果 P1 大于 10 或 P2 = 2，程序段重复停止。

指令 $CONTINUE 开始下一个重复，甚至在当前的一个没有完成时也是如此。$CONTINUE 之后直到 $ENDFOR 编写的程序段，在这次重复中被忽略 。

16.2.5 条件程序段重复 ($WHILE)

$WHILE < 条件 >... $ENDWHILE

该指令中定义下列参数：

< 条件 >　　可以是两个数字、参数或结果是数字的表达式之间的比较。

当条件为真时，执行包含在 $WHILE 和 $ENDWHILE 之间的程序段。在每次新重复的开始之前判断条件。

```
...
N20 $WHILE P1<= 10
N30 P1=P1+1
N40...
N50...
N60  $ENDWHILE
...
```

当 P1 小于或等于 10 时，执行程序段 N30 到 N50。

指令 $BREAK 使结尾程序段重复，当停止条件不满足。程序的执行在 $ENDWHILE 之后的程序段继续。

```
...
N20 $WHILE P1<= 10
N30...
N40 $IF P2==2
N50 $BREAK
N60 $ENDIF
N70...
N80 $ENDWHILE
...
```

程序段停止， 如果 P1 大于 10, 或 P2 = 2。

指令 $CONTINUE 开始下一个重复，甚至在当前一个没有完成时。$CONTINUE 之后直到 $ENDWHILE 编写的程序段， 在这次重复中被忽略。

```
...
N20 $WHILE P1<= 10
 N30...
 N40 $IF P0==2
 N50 $CONTINUE
 N60 $ENDIF
 N70...
 N80...
N80 $ENDWHILE
...
```

如果 P0=2, 忽略程序段 N70 到 N80， 在 N20 开始新的重复。

16.2.6 条件程序段重复 ($DO)

$DO ... $ENDDO < 条件 >

该指令中定义下列参数：

< 条件 > 可以是两个数字、参数或结果是数字的表达式之间的比较。

当条件为真时，重复包含在 $DO 和 $ENDDO 之间的程序段的执行。在每次重复的结尾判断条件，因此至少执行一次程序段组。

```
...
N20 $DO
 N30 P1=P1+1
 N40...
 N50...
N60 $ENDDO P1<=10
N70...
```

当 P1 小于或等于 10 时，执行程序段 N30 到 N50 。

指令 $BREAK 让结尾程序段重复，当停止条件不满足。执行在程序段 $ENDDO 之后继续。

```
...
N20 $DO
 N30...
 N40 $IF P2==2
 N50 $BREAK
 N60 $ENDIF
 N70...
N80 $ENDDO P1<= 10
...
```

如果 P1 大于 10 或 P2 = 2，程序段重复停止。

指令 $CONTINUE 开始下一个 重复，甚至在当前一个没有完成时。$CONTINUE 之后直到 $ENDDO 编写的程序段，在这次重复中被忽略。

```
...
N20 $DO
 N30...
 N40 $IF P0==2
 N50 $CONTINUE
 N60 $ENDIF
 N70...
 N80...
N80 $ENDDO P1<= 10
...
```

如果 P0=2, 忽略程序段 N70 到 N80，在 N20 开始新的重复。

探测固定循环

17

CNC 提供如下探测固定循环。

- 刀具半径和长度标定固定循环。
- 探针标定固定循环。
- 表面测量固定循环。
- 外拐角测量固定循环。
- 内拐角测量杜定循环。
- 角度测量固定循环。
- 拐角和角度测量固定循环。
- 孔测量固定循环。
- 凸台固定循环 。

刀具和探针标定固定循环在 G17, G18 和 G19 平面执行。其它固定循环也可以在功能 G20 定义的任何平面执行。

编程

固定循环用指令 #PROBE 编程。

指令#PROBE调用一个用编码或结果为一个数字的任何表达式表示的探测循环 。同样，它允许使用赋值指令的值来初始化循环的参数。

当使用不止一个探针时，在执行固定循环之前必须选择使用的探针。用指令 #SELECT PROBE 执行选择。(“附加编程指令”)。

注意事项

探测循环是非模态的；因此，每次执行时必须编写。

当执行这些固定循环时使用探针：

- 探针定位在机床一个固定位置，用于标定刀具。
- 定位在安装在主轴上的刀具位置；它将作为一个刀具被对待，将用在各种测量循环中。

执行探测循环不改变先前的 "G" 功能记录，除了刀具半径补偿功能 G41 和 G42。



17.1 #PROBE 1. 刀具标定

用于标定主轴刀具的长度和半径。下列操作是可以使用该循环的。

- 刀具长度标定。
- 刀具长度磨损测量。
- 刀具半径标定。
- 刀具半径磨损测量。
- 刀具半径和长度标定。
- 刀具半径磨损和长度磨损测量。

需要一个桌面探针，安装在机床的某个固定位置，它的边平行于 X, Y 和 Z 轴。

如果是第一次标定刀具，推荐在刀具偏置表中输入估计尺寸。一旦循环结束，刀具表被更新，用当前选择的相应刀具偏置数据。

编程

该循环的编程格式：

#PROBE 1 B I J F K L D S M C N X U Y V Z W

根据执行的操作，不需要定义所有的参数。

参数 X, U, Y, V, Z, W

定义探针位置。它们是可选参数，通常不需要定义。

- 参数 X-Y-Z 指的分别是在第一轴、第二轴和垂直于平面轴的探针的最小坐标。
- 参数 U-V-W 指的分别是在第一轴、第二轴和垂直于平面轴的探针的最大坐标。

在某些机床上，由于在探针的机械定位缺乏重复性，在每次标定前探针必须再次标定 。

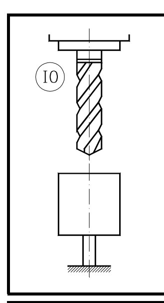
代替每次探针被标定时再定义机床参数，这些坐标可以在这些参数中确定。数控系统不修改机床参数，仅在标定期间考虑在 X, U, Y, V, Z, W 确定的坐标。

如果 X, U, Y, V, Z, W 组的任一个被遗漏，数控系统采用对应机床参数所赋的值。



17.1.1 测量或标定刀具长度

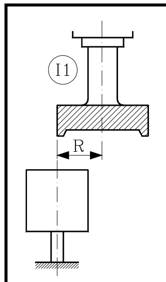
可以对刀杆或刀尖进行标定或测量。当调用固定循环时选择标定或测量的类型 。



关于刀杆

用于钻削刀具、球面铣刀或尺寸小于探针表面积的刀具。

用停止的主轴来执行它。



关于刀尖

用于有多个切削刃的标定刀具 （结束铣削）或尺寸大于探针表面积的刀具。

用停止的主轴或顺铣执行它。

编程

根据执行的操作类型决定编程格式。

- 在刀杆刀具长度标定：

  #PROBE 1 B I0 J0 F X U Y V Z W

- 在末端刀具长度标定：

  #PROBE 1 B I1 J0 F D S N X U Y V Z W

- 在刀杆刀具长度磨损测量：

  #PROBE 1 B I0 J1 F L C X U Y V Z W

- 在末端刀具长度磨损测量：

  #PROBE 1 B I1 J1 F L D S C N X U Y V Z W

B　　安全距离。必须用大于 0 的正值编写。

I　　测量或标定类型。

值	含义
0	刀杆长度
1	刀尖长度
2	测量或标定刀具半径
3	刀具长度和半径

如果没有编写，固定循环采用 "I0" 的值。

J　　执行的操作。

值	含义
0	刀具标定
1	磨损测量

如果没有编写，固定循环采用 "J0" 的值。



F 探测速度。

L 最大允许长度磨损。
如果没有编写，循环采用 "L0" 值。(刀具不会因为长度磨损被拒绝)。

D 探测位置的相关刀具半径和距离。
如果没有编写，循环采用刀具半径值。

S 刀具方向和旋转速度。所选方向必须和切削方向相反 (如果 M3 是 正向，如果 M4 是负向)。
如果没有编写，循环采用 "S0" 值 (主轴停止时标定)。

C 当超过最大允许磨损时的动作。

值	含义
0	显示拒绝刀具和停止循环的信息。
1	循环用同组中的另一把刀具刀具替换刀具。

如果没有编写，固定循环采用 "C0" 值。

N 要测量的切削刃编号，"S" 参数必须用非零值定义。
如果没有编写，循环采用 "N0" 值 (单次测量)。

X, U, Y, V, Z, W
可选参数。

固定循环结束之后

一旦标定循环结束

更新全局算数参数 P299 和在刀具表选择的赋予刀具偏置的值 。

P299 (测量长度) - (先前长度 (L+LW)).

L 测量长度

LW 0

如果要求每个刃的尺寸 (参数 "N")，值将被赋给全局参数 P271 和后面的参数。

一旦磨损测量循环结束

比较测量值和刀具表中所赋理论长度值。

- 如果超过最大允许磨损量，设置 " 已耗刀具 " 指示器，动作如下：

  C0 发出 " 被拒绝刀具 " 信息，中断执行，用户可以选择另一把刀具。

  C1 循环用同组中另一把刀具更换掉刀具。

- 如果测量差值没有超出最大允许值，更新全局算数参数 P299 和在刀具表选择的赋予刀具偏置的值 。

  P299 测量长度 - 理论长度 (L)。

  L 理论长度 (保持先前的值)。

  LW 测量长度 - 理论长度 (L).

如果要求每个刃的尺寸 (参数 "N")，值将被赋给全局参数 P271 和后面的参数。

17.1.2 测量或标定刀具半径

停止的主轴或顺铣时它可以执行。

编程

根据执行的操作类型决定编程格式。

- 刀具半径标定：

  #PROBE 1 B I2 J0 F K S N X U Y V Z W

- 半径磨损测量：

  #PROBE 1 B I2 J1 F K S M C N X U Y V Z W

B 安全距离。必须用大于 0 的正值编写。

I 测量或标定类型。

值	含义
0	刀杆长度
1	刀尖长度
2	测量或标定刀具半径
3	刀具长度和半径

如果没有编写，固定循环采用 "I0" 值。

J 执行操作。

值	含义
0	刀具标定
1	磨损测量

如果没有编写，固定循环采用 "J0" 值。

F 探测进给率

K 探针使用边

值	含义
0	在 X+ 边
1	在 X- 边
2	在 Y+ 边
3	在 Y- 边

如果没有编写，固定循环采用 "K0" 值。

S 刀具方向和旋转速度。所选方向必须和切削方向相反 (如果 M3 是 正向，如果 M4 是负向)。
如果没有编写，循环采用 "S0" 值 (主轴停止时标定)。

M 最大允许半径磨损。
如果没有编写，循环采用 "M0" 值 (刀具不会因为半径磨损被拒绝)。



C　　当超过最大允许磨损时的动作。

值	含义
0	显示拒绝刀具和停止循环的信息。
1	循环用同组中的另一把刀具刀具替换刀具。

如果没有编写，固定循环采用 "C0" 值。

N　　要测量的切削刃编号，"S" 参数 必须用非零值定义。
如果没有编写，循环采用 "N0" 值 (单次测量)。

X, U, Y, V, Z, W
可选参数。

循环结束后

一旦标定循环结束

更新全局算数参数 P298 和在刀具表选择的赋予刀具偏置的值。

P298	(测量半径) - (先前半径 (R+RW))。
R	测量半径。
RW	0

如果要求每个刃的尺寸 (参数 "N")，值将被赋给全局参数 P251 和后面的参数。

一旦磨损测量循环结束

比较测量值和刀具表中所赋理论半径值。

- 如果超过最大允许磨损量，设置 " 已耗刀具 " 指示器，动作如下：

C0	发出 " 被拒绝刀具 " 信息，中断执行，用户可以选择另一把刀具。
C1	循环用同组中另一把刀具更换掉刀具。

- 如果测量差值没有超出最大允许值，更新全局算数参数 P298 和在刀具表选择的赋予刀具偏置的值 。

P298	测量半径 - 理论半径 (R)。
R	理论半径 (保持先前的值)。
RW	测量半径 - 理论半径 (R)。

如果要求每个刃的尺寸 (参数 "N")，值将被赋给全局参数 P251 和后面的参数。



17.1.3 测量或标定刀具半径和长度

停止的主轴或顺铣时它可以执行。

编程

根据执行的操作类型决定编程格式。

- 刀具半径标定 :

  #PROBE 1 B I3 J0 F K D S N X U Y V Z W

- 半径磨损测量 :

  #PROBE 1 B I3 J1 F K L D S M C N X U Y V Z W

B 安全距离。必须用大于 0 的正值编写。

I 测量或标定类型。

值	含义
0	刀杆长度
1	刀尖长度
2	测量或标定刀具半径
3	刀具长度和半径

如果没有编写，固定循环采用 "I0" 值。

J 执行操作。

值	含义
0	刀具标定
1	磨损测量

如果没有编写，固定循环采用 "J0" 值。

F 探测进给率

K 探针使用边

值	含义
0	在 X+ 边
1	在 X- 边
2	在 Y+ 边
3	在 Y- 边

如果没有编写，固定循环采用 "K0" 值。

L 最大允许长度磨损。
如果没有编写，循环采用 "L0" 值。(刀具不会因为长度磨损被拒绝).

D 相关被探测刀杆的半径或距离。
如果没有编写，循环采用刀具半径值。

S 刀具方向和旋转速度。所选方向必须和切削方向相反 (如果 M3 是 正向，如果 M4 是负向)。
如果没有编写，循环采用 "S0" 值 (主轴停止时标定)。



M 最大允许半径磨损。
如果没有编写，循环采用 "M0" 值 (刀具不会因为半径磨损被拒绝)。

C 当超过最大允许磨损时的动作。

值	含义
0	显示拒绝刀具和停止循环的信息。
1	循环用同组中的另一把刀具刀具替换刀具。

如果没有编写，固定循环采用 "C0" 值。

N 要测量的切削刃编号，"S" 参数 必须用非零值定义。
如果没有编写，循环采用 "N0" 值 (单次测量)。

X, U, Y, V, Z, W
可选参数。

循环结束后

一旦标定循环结束

更新全局算数参数 P298， P299 和在刀具表选择的赋予刀具偏置的值。

P298	测量半径 - 先前半径 (R+RW)
P299	测量长度 - 先前长度 (L+LW)
R	测量半径
L	测量长度
RW	0
LW	0

如果要求每个刃的尺寸 (参数 "N")，长度将被赋给全局算数参数 P271 和后面的参数，半径将被赋给全局算数参数 P251 和后面的参数。

一旦磨损测量循环结束

比较测量值和刀具表中所赋理论值。

- 如果超过最大允许磨损量，设置 " 已耗刀具 " 指示器，动作如下 :

C0	发出 " 被拒绝刀具 " 信息，中断执行，用户可以选择另一把刀具。
C1	循环用同组中另一把刀具更换掉刀具。

- 如果测量差值没有超出最大允许值，更新全局算数参数 P299 和在刀具表选择的赋予刀具偏置的值。

P299	测量长度 - 理论长度 (L)
L	理论长度 (保持先前值)
LW	测量长度 - 理论长度 (L)

比较测量值和刀具表中所赋理论值。

- 如果超过最大允许磨损量，设置 " 已耗刀具 " 指示器，动作如下 :

C0	发出 " 被拒绝刀具 " 信息，中断执行，用户可以选择另一把刀具。
C1	循环用同组中另一把刀具更换掉刀具。

- 如果测量差值没有超出最大允许值，更新全局算数参数 P298 和在刀具表选择的赋予刀具偏置的值。

P298	测量半径 - 理论半径 (R)
R	理论半径 (保持先前值)
RW	测量半径 - 理论半径 (R)

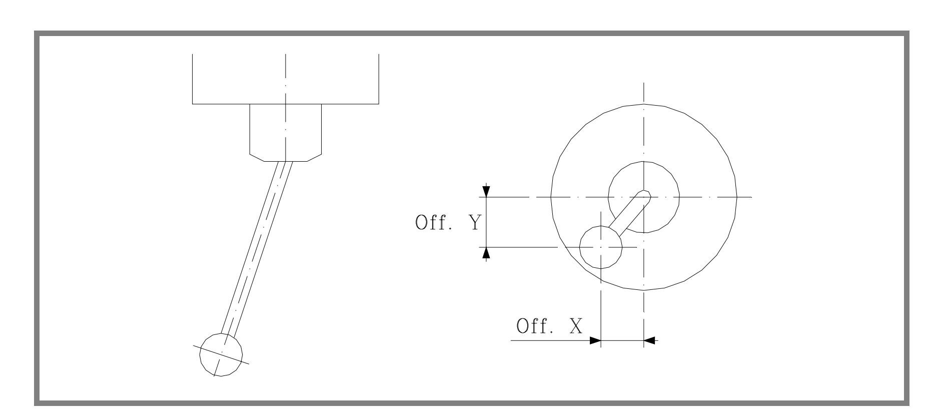
如果要求每个刃的尺寸 (参数 "N")，长度将被赋给全局算数参数 P271 和后面的参数，半径将被赋给全局算数参数 P251 和后面的参数。



17.2 #PROBE 2. 探针标定

用于标定安装在主轴上的探针。该探针必须在前面标定了长度，将用于探测固定循环。

该循环利用前面加工的已知中心和尺寸的孔测量探针球的中心轴和刀具安装轴之间的偏差。



数控系统将每个测量探针当作一个刀具。对应于每个探针的刀具偏置域有下列含义 :

R	探针球的半径。该数值将手动加载到表中。
L	探针长度。该数值将由刀具长度标定循环指定。
Off. X	沿横坐标轴探针的球心与刀具安装孔的轴之间的差值。由循环给出。
Off. Y	沿纵坐标轴探针的球心与刀具安装孔的轴之间的差值。由循环给出。

按下列步骤进行标定 :

1. 一旦知道了探针的特性，探针球的半径值 (R) 将可以手动输入相应的刀具偏置。
2. 在选择相应的刀具号和刀具偏置后，将完成刀具长度标定循环。数值 "L" 将被更新，"Off. Z" 将被初始化为 0。
3. 执行探针标定固定循环，更新 "Off. X" 和 "Off. Y" 的数值。



编程

循环编程格式如下 :

#PROBE 2 X Y Z B J E H F

X 沿横坐标轴孔中心的实际坐标。

Y 沿纵坐标轴孔中心的实际坐标。

Z 沿垂直于平面的轴孔中心的实际坐标。

B 定义安全距离。必须用大于 0 的正值编写。

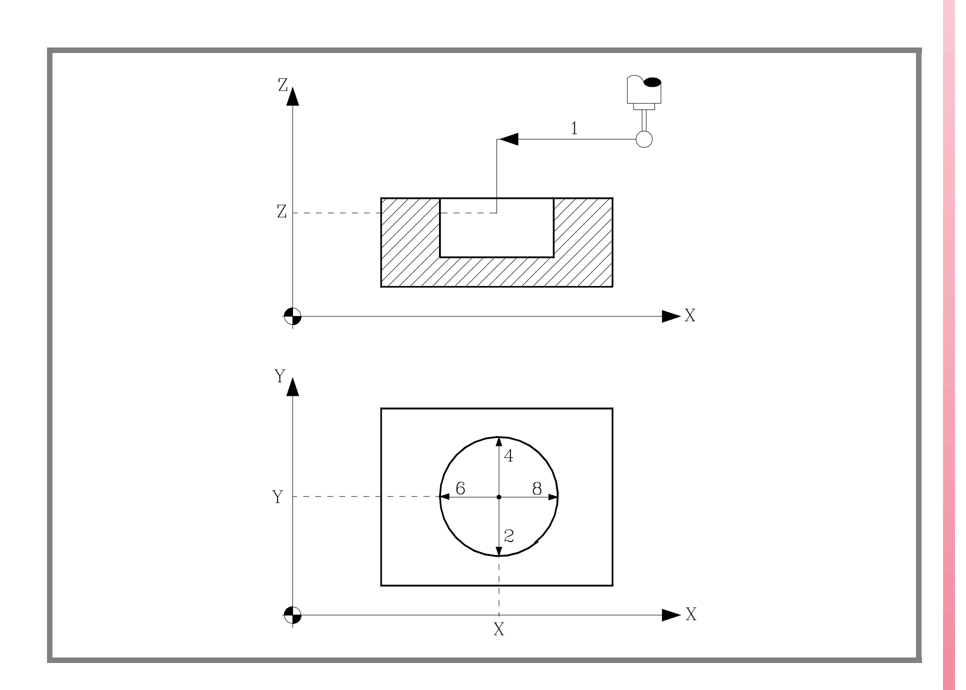
J 定义孔的实际直径。必须用大于 0 的正值编写。

E 定义探针在初次探测后运动回的距离。 必须用大于 0 的正值编写。

H 定义初次测量的进给率。

F 定义探测进给率。

基本操作



1. 接近运动。

探针快速 (G00) 从循环调用点运动到孔的中心 。

接近运动分两个阶段完成 :

·1· 在主工作平面的运动。

·2· 沿垂直轴的运动。



2. 探测运动

探测运动由下列阶段组成 :

·1· 探针以进给率 H 沿纵坐标运动，直到接收到探针信号。

每次探测运动的最大距离为 "B+(J/2)", 如果超过了该距离数控系统仍然没有接收到探针信号，将发送错误信息并将轴的运动停止。

·2· 探针快速 (G00) 返回到指定距离 (E)。

·3· 探针沿垂直轴以指定的进给率 (F) 运动，直到接收到探针信号。

3. 回退运动

探针快速 (G00) 从探测位置回退到孔的实际中心点。

4. 第二次探测运动。

与上面相同。

5. 回退运动

探针沿纵坐标轴快速 (G00) 从探测位置回退到孔的实际中心点。

6. 第三次探测运动

与上面相同。

7. 回退运动

探针快速 (G00) 从测量位置回退到孔的实际中心点。

8. 第四次探测运动

与上面相同。

9. 回退运动

该运动由下列运动组成 :

·1· 探针快速 (G00) 从探测位置回退到孔的实际中心点。

·2· 沿垂直轴运动到循环调用点的坐标。

·3· 在主平面运动到循环调用点。

一旦循环结束，数控系统将更新在刀具偏置表中所选择的此时刀具偏置值 "Off. X" 和 "Off. Y" 。

另一方面，算数参数 P298 和 P299，返回关于横坐标轴和纵坐标轴的赋予轴机床参数 PROBEDELAY 的最佳值。



17.3 #PROBE 3. 表面测量固定循环

用安装在主轴上的探针，该探针必须在前面用表定循环 #PROBE 1 和 #PROBE 进行了标定。

使用该循环，可以修正用于加工表面的刀具的刀具偏置。这种修正只有在测量误差超过编程值时发生 。

编程

循环可以在工作平面编写。根据工作平面，循环的理论坐标可以通过几种方式定义:

- 在有效工作平面，除了由轴 A-B-C 形成的平面。

#PROBE 3 X50 Y65 Z15 …	主要轴 X-Y-Z
#PROBE 3 X1=50 Y2=65 Z1=15 …	主要轴 X1-Y2-Z1

- 使用参数　X-Y-Z。当平面不是由这些轴形成的时，这些参数被认为分别是第一轴、第二轴和垂直于平面的轴的坐标。

#PROBE 3 X50 Y65 Z15 …	主要轴 X1-B-C

在 G17, G18 G19 平面的编程格式 :

```
#PROBE 3 X Y Z B K F C D L
```

X　要求测量点上的 X 理论坐标。

Y　要求测量点上的 Y 理论坐标。

Z　要求测量点上的 Z 理论坐标。

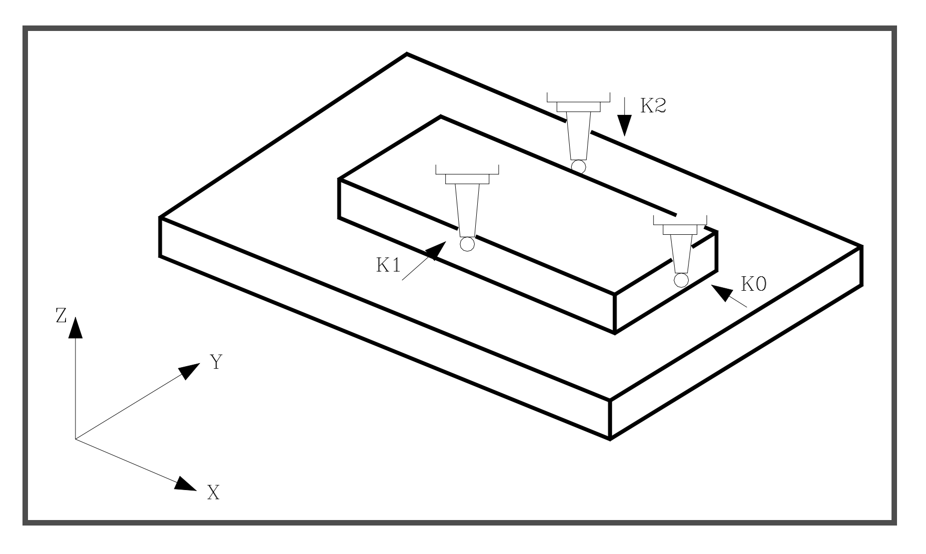
B　定义安全距离。必须用大于 0 的正数编写。
当调用循环时，探针安装的位置相对于测量点必须大于该值。

K　定义测量表面需要的轴。

值	测量用轴
0	用工作平面的横坐标轴。
1	用工作平面的纵坐标轴。
2	用工作平面的垂直轴。 .

如果没有编写，循环采用 "K0" 值。





F　　定义探测进给率。

C　　指定循环必须结束位置。

值	含义
0	探针返回到循环调用点。
1	循环在测量点结束。垂直轴返回到循环调用点。

如果没有编写，循环采用 "C0" 值。

T　　校正偏置的刀具。
如果没有编写，数控系统认为是用于加工的刀具。

D　　定义测量循环一旦结束要修正的刀具偏置号。
如果没有编写或编写 0 值，数控系统认为不需要修正。

L　　定义测量误差的允许公差。将用绝对值编写，只有当误差超过该数值时才对刀具偏置进行修正。
如果没有编写，数控系统设置该参数为 "0"。

测量结果

一旦循环结束，数控系统返回在测量中获得的下列普通算数参数的实际值：

P298　　实际表面坐标。

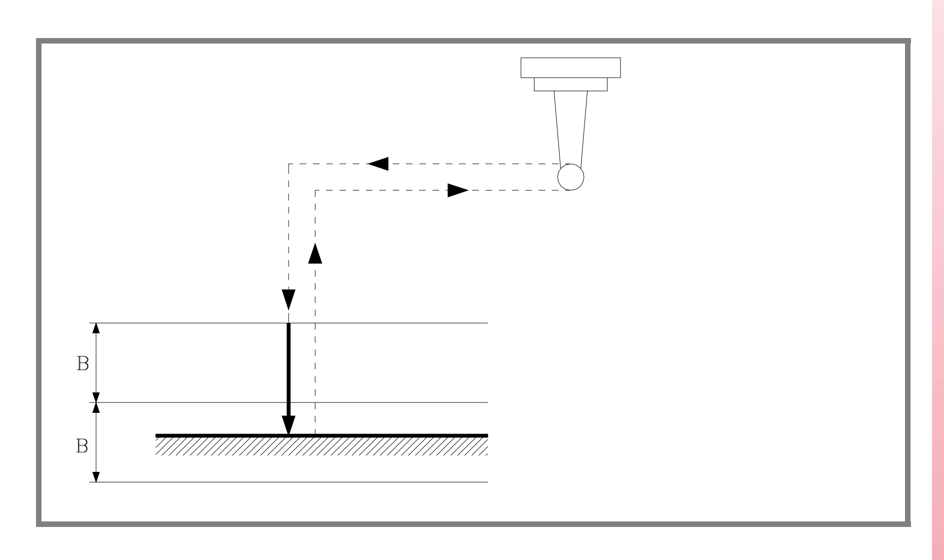
P299　　D 探测错误。表面的实际坐标和编程理论坐标之间的差值。

如果选择了刀具偏置号 (D)，在测量误差等于或大于公差 (L) 时，数控系统将修改该刀具偏置值。

依据用于测量的轴，将应用修正在长度值或半径值。

- 如果测量用垂直轴在工作平面完成，将修改指定刀具偏置 (D) 的长度磨损 (LW)。
- 如果测量用组成平面的轴完成，将修改制定刀具偏置 (D) 的半径磨损 (RW) 。

基本操作



1. 接近运动

探针快速 (G00) 从循环调用点运动到接近点。

该点在被测点的前面，沿探测轴 (K) 的安全距离为 (B) 。

接近运动分两阶段完成 :

·1· 在主工作平面的运动。

·2· 沿垂直轴的运动。

2. 探测运动

探针沿选定的轴 (K) 以指定的进给率 (F) 运动，直到接到探针信号。

每次探测运动的最大距离为 2B。如果超过了该距离，数控系统仍然没有接收到探针的信号，将发送错误信息并将轴的运动停止。

一旦探测完成，数控系统将把接受到探针信号时轴的实际位置作为理论位置。



3. 退回运动

探针快速 (G00) 从探测位置回退到循环调用点。

退回运动分三阶段完成 :

·1· 沿探测轴运动到接近点。

·2· 沿垂直轴运动到循环调用点。

·3· 当编写了 (C0)，在主工作平面运动到循环调用点。



17.4 #PROBE 4. 外拐角测量固定循环

用安装在主轴上的探针。该探针必须在前面用标定固定循环 #PROBE 1 和 #PROBE 2 进行了标定。

编程

循环可以在任意工作平面编写。根据工作平面，循环的理论坐标可以通过几种方式定义：

- 在有效工作平面，除了是由轴 A-B-C 形成的平面。

#PROBE 4 X50 Y65 Z15 ...	主要轴 X-Y-Z
#PROBE 4 X1=50 Y2=65 Z1=15 ...	主要轴 X1-Y2-Z1

- 使用参数 X-Y-Z。当平面不是由这些轴形成的时，这些参数被认为分别是第一轴、第二轴和垂直于平面的轴的坐标。

#PROBE 4 X50 Y65 Z15 ...	主要轴 X1-B-C

在 G17, G18 或 G19 平面的编程格式：

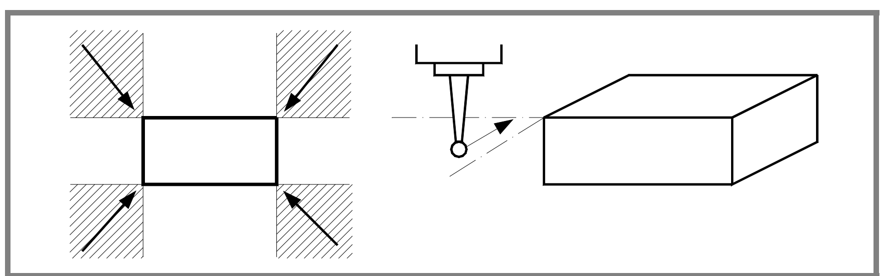
```
#PROBE 4 X Y Z B F
```

X 要求测量点上的 X 理论坐标。

Y 要求测量点上的 Y 理论坐标。

Z 要求测量点上的 Z 理论坐标。

根据被测工件拐角，在调用循环之前，必须把探针放置在对应的阴影区域，如图所示。



B 定义安全距离。必须用大于 0 的正数编写。
当调用循环时，探针安装的位置相对于测量点必须大于该值。

F 探测进给率。

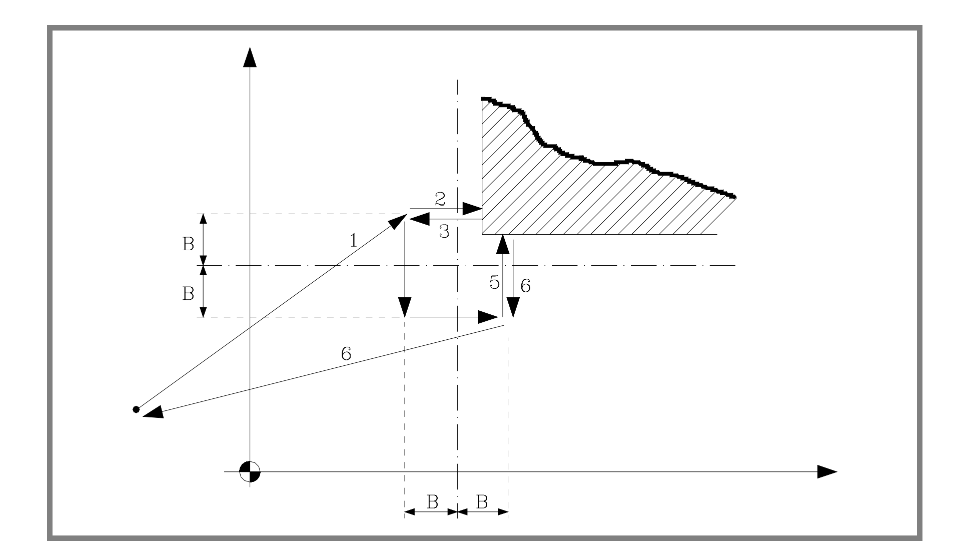
测量结果

一旦循环结束，数控系统返回在测量中获得的下列普通算数参数的实际值：

P296	拐角沿横坐标的实际坐标。
P297	拐角沿纵坐标的实际坐标。
P298	沿横坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。
P299	沿纵坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。



基本操作



1. 接近运动

探针快速 (G00) 从循环调用点运动到第一接近点，与被测面点距离 (B) 。

接近运动分两阶段完成 :

·1· 在主工作平面的运动。

·2· 沿垂直轴的运动。

2. 探测运动

探针沿横坐标轴以指定的进给率 (F) 运动，直到接到探针信号。

每次探测运动的最大距离为 2B。如果超过了该距离，数控系统仍然没有接收到探针的信号，将发送错误信息并将轴的运动停止。

3. 退回运动

探针快速 (G00) 从探测位置回退到第一接近点。



4. 第二接近运动

探针快速 (G00) 从第一接近运动到第二接近点 。

该接近运动分两阶段完成 :

·1· 沿纵坐标轴运动。

·2· 沿横坐标轴运动。

5. 第二次探测运动

沿纵坐标轴以指定的进给率 (F) 运动，直到接到探针信号。

每次探测运动的最大距离为 2B。如果超过了该距离数控系统仍然没有接收到探针的信号，将发送错误信息并将轴的运动停止。

6. 退回运动

探针快速 R(G00) 从第二探测点运动到循环调用点。

该退回运动分三个阶段完成 :

·1· 沿探测轴运动到第二接近点。

·2· 沿垂直轴运动到该轴的循环调用位置。

·3· 在主工作面运动到循环调用点。



17.5 #PROBE 5. 内拐角测量固定循环

用安装在主轴上的探针。该探针必须在前面用标定固定循环 #PROBE 1 和 #PROBE 2 进行了标定。

编程

循环可以在任意工作平面编写。根据工作平面，循环的理论坐标可以通过几种方式定义：

- 在有效工作平面，除了由轴 A-B-C 形成的平面。

#PROBE 5 X50 Y65 Z15 ...	主要轴 X-Y-Z
#PROBE 5 X1=50 Y2=65 Z1=15 ...	主要轴 X1-Y2-Z1

- 使用参数 X-Y-Z。当平面不是由这些轴形成的时，这些参数被认为分别是第一轴、第二轴和垂直于平面的轴的坐标。

#PROBE 5 X50 Y65 Z15 ...	主要轴 X1-B-C

在 G17, G18 或 G19 平面的编程格式 :

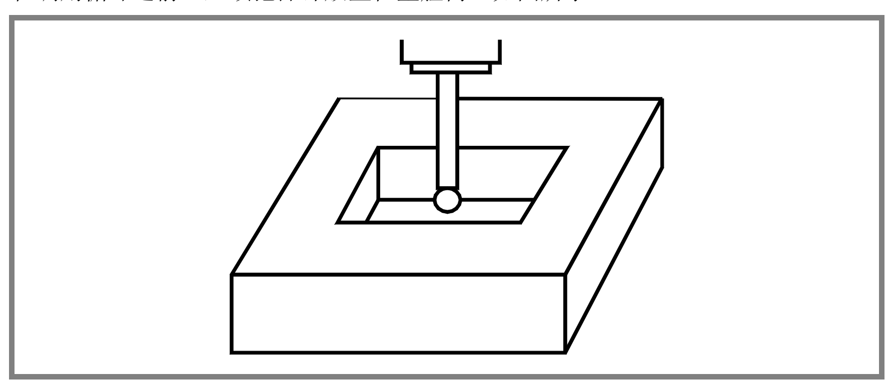
```
#PROBE 5 X Y Z B F
```

X 沿 X 轴要测量拐角的理论坐标。

Y 沿 Y 轴要测量拐角的理论坐标。

Z 沿 Z 轴要测量拐角的理论坐标。

在调用循环之前，必须把探针放置在型腔内，如图所示。



B 定义安全距离。必须用大于 0 的正数编写。
当调用循环时，探针安装的位置相对于测量点必须大于该值。

F 探测进给率。

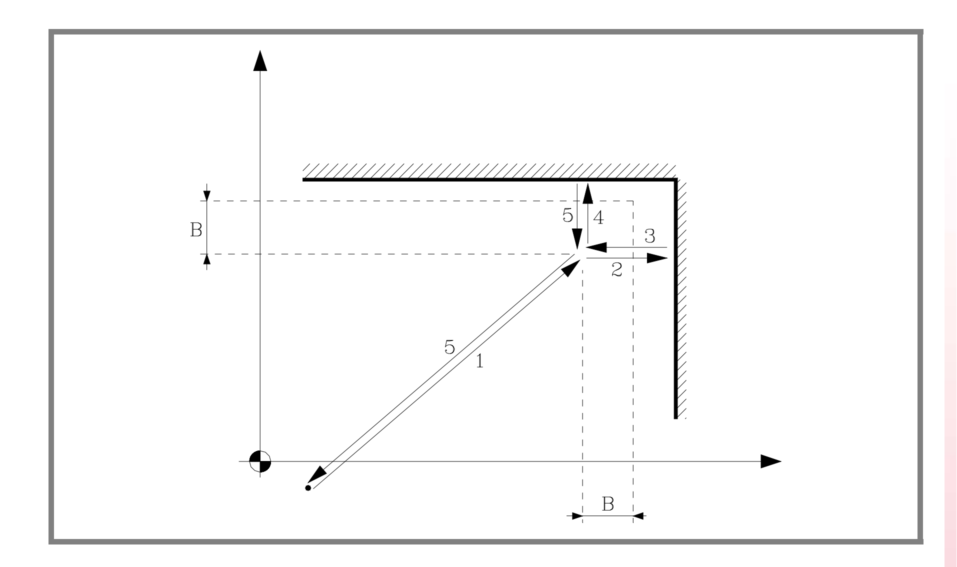
测量结果

一旦循环结束，数控系统返回在测量中获得的下列普通算数参数的实际值：

P296	拐角沿横坐标的实际坐标。
P297	拐角沿纵坐标的实际坐标。
P298	沿横坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。
P299	沿纵坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。



基本操作



1. 接近运动

探针快速 (G00) 从循环调用点运动到距离两被测面距离为 (B) 的第一接近点。

接近运动分两阶段完成 :

·1· 在主工作平面的运动。

·2· 沿垂直轴的运动。

2. 探测运动

探针沿横坐标轴以指定的进给率 (F) 运动，直到接到探针信号。

每次探测运动的最大距离为 2B。如果超过了该距离，数控系统仍然没有接收到探针的信号，将发送错误信息并将轴的运动停止。

3. 退回运动

探针快速 (G00) 从探测点运动到接近点 。

4. 第二次探测

探针沿纵坐标以指定的进给率 (F) 运动，直到接到探针信号。

每次探测运动的最大距离为 2B。如果超过了该距离数控系统仍然没有接收到探针的信号，将发送错误信息并将轴的运动停止。



5. 退回运动

探针快速 (G00) 从第二次的探测点运动到循环调用点。

该退回运动分三阶段完成 :

·1· 沿探测轴运动到接近点。

·2· 沿垂直轴运动到该轴的循环调用点位置。

·3· 在主工作面运动到循环调用点。



17.6 #PROBE 6. 角度测量固定循环

用安装在主轴上的探针。该探针必须在前面用标定固定循环 #PROBE 1 和 #PROBE 2 进行了标定。

该循环可以用于测量 ±45º 之间的角。

- 如果被测的角度 >=45º, 数控系统将显示相应的错误。
- 如果被测得角度 <=–45º, 探针将与工件碰撞 。

编程

循环可以在任意工作平面编写。根据工作平面，循环的理论坐标可以通过几种方式定义：

- 在有效工作平面，除了由轴 A-B-C 形成的平面。

#PROBE 6 X50 Y65 Z15 …	主要轴 X-Y-Z
#PROBE 6 X1=50 Y2=65 Z1=15 …	主要轴 X1-Y2-Z1

- 使用参数 X-Y-Z。当平面不是由这些轴形成的时，这些参数被认为分别是第一轴、第二轴和垂直于平面的轴的坐标。

#PROBE 6 X50 Y65 Z15 …	主要轴 X1-B-C

在 G17, G18 或 G19 平面的编程格式 :

```
#PROBE 6 X Y Z B F
```

X 沿 X 轴要测角度的理论坐标。

Y 沿 Y 轴要测角度的理论坐标。

Z 沿 Z 轴要测角度的理论坐标。

B 定义安全距离。必须用大于 0 的正数编写。
当调用循环时，探针必须安装在相对测量点大于两倍该值的位置。

F 探测进给率。

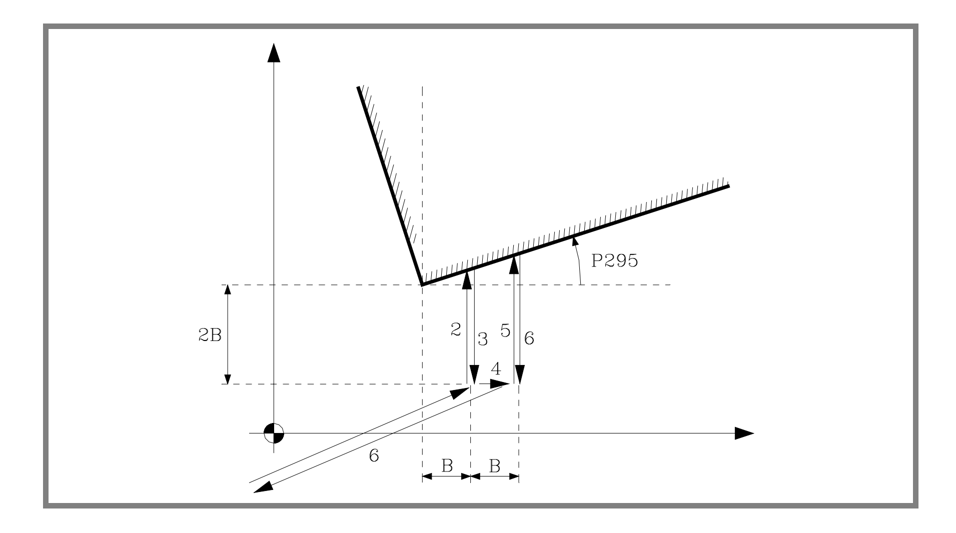
测量结果

一旦循环结束，数控系统返回在测量中获得的下列普通算数参数的实际值 :

P295 工件相对横坐标的倾斜角度。



基本操作



1. 接近运动

   探针快速 (G00) 从循环调用点运动到距离编写的顶点为 (B) 的第一接近点，距离探测的端面距离为 (2B)。

   接近运动分两个阶段完成 :

   ·1· 在主工作面的运动。

   ·2· 沿垂直轴的运动。

2. 探测运动

   探针沿纵坐标轴以指定的进给率 (F) 运动，直到接到探针信号。

   最大探测距离是 3B。如果一旦达到该距离，但是数控系统没有收到探针信号，它将发送相关错误代码并停止轴的运动。

3. 退回运动

   探针快速 (G00) 从探测点运动到第一接近点 。

4. 第二次接近运动

   探针快速 (G00) 从第一接近点运动到第二接近点。距离第一接近点距离为 (B) 。

5. 第二次探测运动

   探针沿纵坐标轴以指定进给率 (F) 运动，直到接到探针信号。

   最大探测距离是 4B。如果一旦达到该距离，但是数控系统没有收到探针信号，它将发送相关错误代码并停止轴的运动。

6. 退回运动

   探针快速 (G00) 从第二次探测点运动到循环调用点。

   该退回运动分三阶段完成 :

   ·1· 沿探针轴运动到第二接近点。

   ·2· 沿垂直轴运动到该轴的循环调用点位置。

   ·3· 在主工作面运动到循环调用点。

17.7 #PROBE 8. 外拐角和角度测量固定循环

用安装在主轴上的探针。该探针必须在前面用标定固定循环 #PROBE 1 和 #PROBE 2 进行了标定。

该循环可以用于测量 ±45º 之间的角。

- 如果被测的角度 >=45º, 数控系统将显示相应的错误。
- 如果被测得角度 <=–45º, 探针将与工件碰撞。

编程

循环可以在任意工作平面编写。根据工作平面，循环的理论坐标可以通过几种方式定义：

- 在有效工作平面，除了由轴 A-B-C 形成的平面。

#PROBE 7 X50 Y65 Z15 …	主要轴 X-Y-Z
#PROBE 7 X1=50 Y2=65 Z1=15 …	主要轴 X1-Y2-Z1

- 使用参数 X-Y-Z。当平面不是由这些轴形成的时，这些参数被认为分别是第一轴、第二轴和垂直于平面的轴的坐标。

#PROBE 7 X50 Y65 Z15 …	主要轴 X1-B-C

在 G17, G18 或 G19 平面的编程格式 :

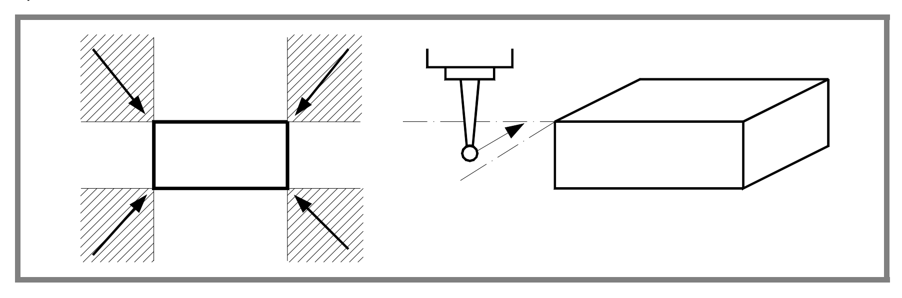
```
#PROBE 7 X Y Z B F
```

X 沿 X 轴要测拐角的理论坐标。

Y 沿 Y 轴要测拐角的理论坐标。

Z 沿 Z 轴要测拐角的理论坐标。

根据被测工件拐角，在调用循环之前，必须把探针放置在对应的阴影区域，如图所示。



B 定义安全距离。必须用大于 0 的正数编写。
当调用循环时，探针必须安装在相对测量点大于两倍该值的位置。

F 探测进给率。

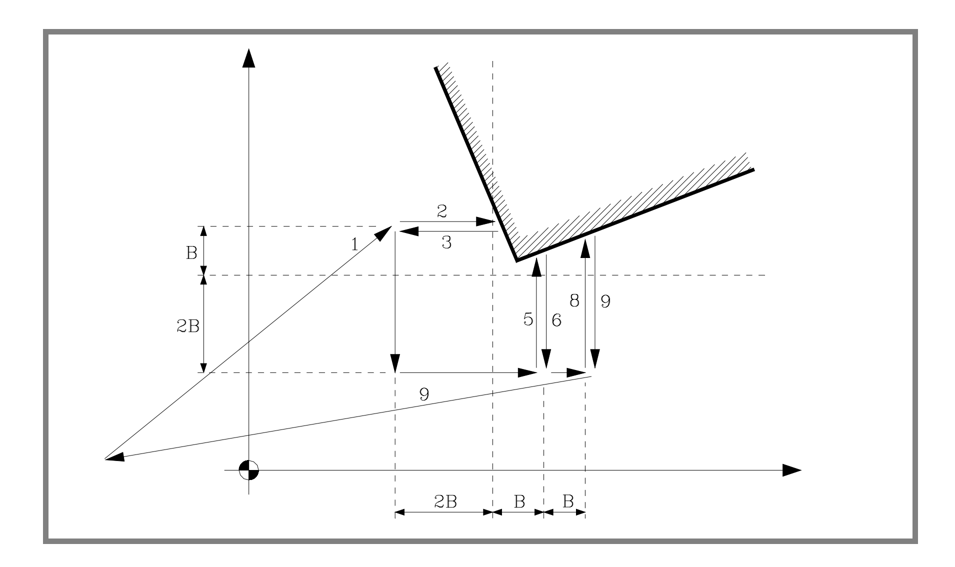


测量结果

一旦循环结束，数控系统返回在测量中获得的下列普通算数参数的实际值 :

P295	工件相对横坐标的倾斜角度。
P296	沿横坐标拐角的实际坐标。
P297	沿纵坐标拐角的实际坐标。
P298	沿横坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。
P299	沿纵坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。

基本操作



1. 接近运动

探针快速 (G00) 从循环调用点运动到距离第一探测面为 (2B) 的第一接近点。

接近运动分两个阶段完成 :

·1· 在主工作面的运动。

·2· 沿垂直轴的运动。

2. 探测运动

探针沿横坐标轴以指定的进给率 (F) 运动，直到接到探针信号。

最大探测距离是 3B。如果一旦达到该距离，但是数控系统没有收到探针信号，它将发送相关错误代码并停止轴的运动。

3. 退回运动

探针快速 (G00) 从探测点运动到第一接近点。

4. 第二接近运动

探针快速 (G00) 从第一接近点运动到第二接近点，距离第二探测面的距离为 (2B) 。

该接近运动分两个阶段完成 :

·1· 沿纵坐标的运动。

·2· 沿横坐标的运动。

5. 第二次探测运动

探针沿纵坐标轴以指定进给率 (F) 运动，直到接到探针信号。

最大探测距离是 3B。如果一旦达到该距离，但是数控系统没有收到探针信号，它将发送相关错误代码并停止轴的运动。

6. 退回运动

探针快速 (G00) 从探测点运动到第二接近点。

7. 第三次接近运动

探针快速 (G00) 从第二接近点运动到第三接近点。距离前一点为 (B) 位置。

8. 第三次探测运动

探针沿纵坐标以指定的进给率 (F) 运动，直到接到探针信号。

最大探测距离是 (4B)，如果一旦达到了该距离，但是数控系统仍然没有接收到探针信号，将发送错误信息并将轴的运动停止。

9. 退回运动

探针快速 (G00) 从第三次探测点运动到循环调用点。

该退回运动分三个阶段完成 :

·1· 沿探测轴运动到第三接近点。

·2· 沿垂直轴运动到该轴的循环调用点位置。

·3· 在主工作面运动到循环调用点。

17.8 #PROBE 8. 孔测量固定循环

用安装在主轴上的探针。该探针必须在前面用标定固定循环 #PROBE 1 和 #PROBE 2 进行了标定。

编程

循环可以在任意工作平面编写。根据工作平面，循环的理论坐标可以通过几种方式定义：

- 在有效工作平面，除了由轴 A-B-C 形成的平面。

#PROBE 8 X50 Y65 Z15 ...	主要轴 X-Y-Z
#PROBE 8 X1=50 Y2=65 Z1=15 ...	主要轴 X1-Y2-Z1

- 使用参数　X-Y-Z。当平面不是由这些轴形成的时，这些参数被认为分别是第一轴、第二轴和垂直于平面的轴的坐标。

#PROBE 8 X50 Y65 Z15 ...	主要轴 X1-B-C

在 G17, G18 或 G19 平面的编程格式 :

#PROBE 8 X Y Z B J E C H F

X　　沿 X 轴孔中心的理论坐标。

Y　　沿 Y 轴孔中心的理论坐标。

Z　　Y 沿 Z 轴孔中心的理论坐标。

B　　定义安全距离。必须用大于 0 的正数编写。

J　　定义孔的理论直径。必须用大于 0 的正数编写。

该循环允许被测量的孔的直径不大于 (J+B)。

E　　定义探针在初次探测后运动回来的距离。必须用大于 0 的正数编写。

C　　确定在何位置探测循环必须结束。

值	含义
0	探针返回循环调用点。
1	循环在实际孔中心处结束。

如果没有编写，固定循环将采用 "C0" 值。

H　　定义第一次探测运动的进给率。

F　　探测进给率。

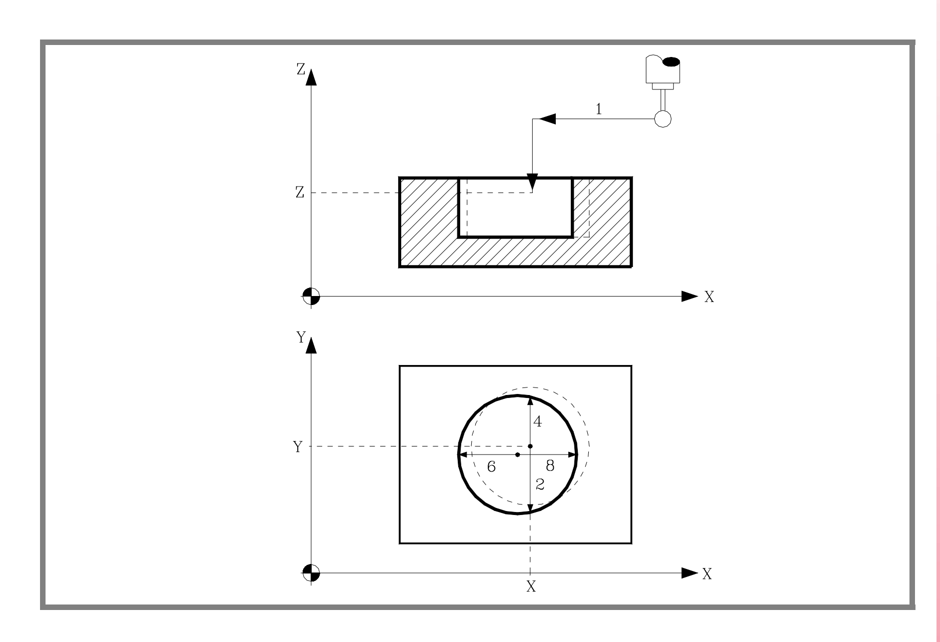


测量结果

一旦循环结束，数控系统返回在测量中获得的下列普通算数参数的实际值 :

P294	孔直径。
P295	孔的直径误差。实际直径和编程直径之间的差值。
P296	沿横坐标孔中心的实际坐标。
P297	沿纵坐标孔中心的实际坐标。
P298	沿横坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。
P299	沿纵坐标检测到的误差。拐角的实际坐标和编程理论坐标之间的差值。

基本操作



1. 接近运动

探针快速 (G00) 从循环调用点运动到孔的中心点。

接近运动分两个阶段完成 :

·1· 在主工作面的运动。

·2· 沿垂直轴的运动。

2. 探测运动

由下列运动组成 :

·1· 探针沿纵坐标轴以指定的进给率 (H) 运动，直到接到探针信号。

最大探测距离是 "B+(J/2)"。如果一旦达到该距离，但是数控系统没有收到探针信号，它将发送相关错误代码并停止轴的运动。

·2· 探针快速 (G00) 返回 (E) 指定的距离。

·3· 探针沿纵坐标轴以指定的进给率 (F) 运动，直到接到探测信号。



3. 退回运动

探针快速 (G00) 从探测点运动到孔的理论中心。

4. 第二次探测运动

与上相同。

5. 退回运动

探针快速 (G00) 沿纵坐标轴从探测点运动到孔的实际中心 (计算的) 。

6. 第三次探测运动

与上相同。

7. 退回运动

探针快速 (G00) 从探测点运动到孔的理论中心 。

8. 第四次探测运动

与上相同。

9. 退回运动

探针快速 (G00) 从探测点运动到孔的理论中心。

当编写 (C0), 探针运动到循环调用点。

·1· 沿垂直轴运动到循环调用点。

·2· 在主工作面运动到循环调用点。



17.9 #PROBE 9. 凸台测量固定循环

用安装在主轴上的探针。该探针必须在前面用标定固定循环 #PROBE 1 和 #PROBE 2 进行了标定。

编程

循环可以在任意工作平面编写。根据工作平面，循环的理论坐标可以通过几种方式定义：

- 在有效工作平面，除了由轴 A-B-C 形成的平面。

#PROBE 9 X50 Y65 Z15 ...	主要轴 X-Y-Z
#PROBE 9 X1=50 Y2=65 Z1=15 ...	主要轴 X1-Y2-Z1

- 使用参数 X-Y-Z。当平面不是由这些轴形成的时，这些参数被认为分别是第一轴、第二轴和垂直于平面的轴的坐标。

#PROBE 9 X50 Y65 Z15 ...	主要轴 X1-B-C

在 G17, G18 或 G19 平面的编程格式 :

```
#PROBE 9 X Y Z B J E C H F
```

X 沿 X 轴凸台中心的理论坐标。

Y 沿 Y 轴凸台中心的理论坐标。

Z 沿 Z 轴凸台中心的理论坐标。

B 安全距离。必须用大于 0 的正数编写。

J 定义理论凸台直径。必须用大于 0 的正数编写。

该循环可用于测量直径不大于 (J+B) 的凸台。

E 定义探针在初次探测后运动回来的距离。必须用大于 0 的正数编写。

C 确定在何位置探测循环必须结束。

值	含义
0	探针返回循环调用点。
1	循环将在探针定位在凸台中心上结束，与编程理论坐标距离 (B) 。

如果没有编写，固定循环将采用 "C0" 值。

H 定义第一次探测运动的进给率。

F 探测进给率。

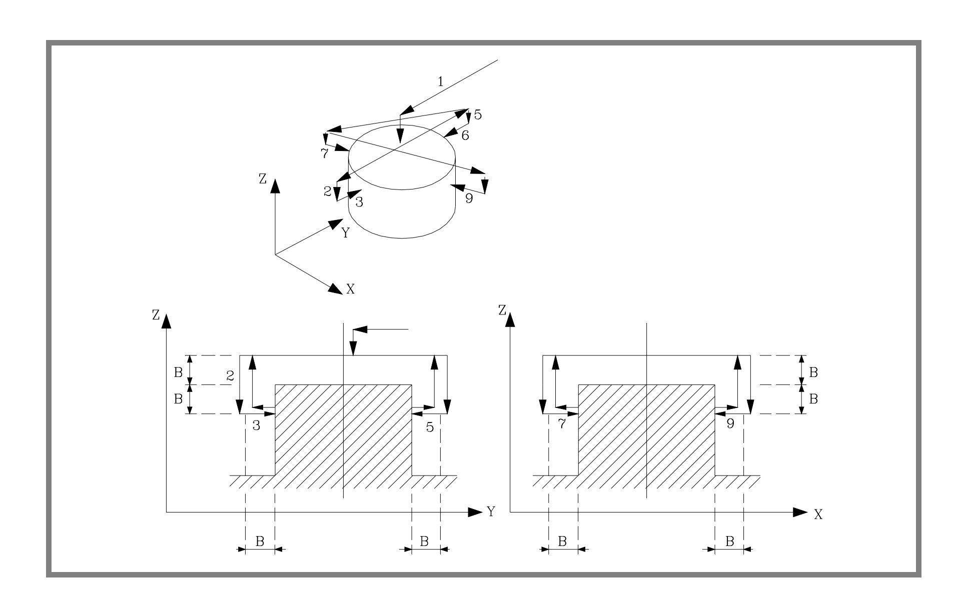


测量结果

一旦循环结束，数控系统返回在测量中获得的下列普通算数参数的实际值 :

P294	凸台直径。
P295	凸台直径误差。直径实际值和编程值之间的差值。
P296	沿横坐标轴凸台中心的实际坐标。
P297	沿纵坐标轴凸台中心的实际坐标。
P298	沿横坐标检测到的误差。凸台中心的实际坐标和编程理论坐标之间的差值。
P299	沿纵坐标检测到的误差。凸台中心的实际坐标和编程理论坐标之间的差值。

基本操作



1. 定位在凸台中心上。

探针快速 (G00) 从循环调用点运动到凸台中心。

该接近运动分两阶段完成 :

·1· 在主工作平面运动。

·2· 沿垂直轴运动到距编程表面 (B)。

2. 运动到第一接近点

探针的快速 (G00) 运动由下列运动组成 :

·1· 沿纵坐标轴的运动。

·2· 沿垂直轴运动距离 (2B)。

3. 探测运动

探测运动由三个阶段组成 :

·1· 探针沿纵坐标轴以指定进给率 (H) 运动，直到接到探针信号。

最大探测距离是 "B+(J/2)"。如果一旦达到该距离，但是数控系统没有收到探针信号，它将发送相关错误代码并停止轴的运动。

·2· 探针快速 (G00) 返回 (E) 指定的距离。

·3· 探针沿纵坐标轴以指定的进给率 (F) 运动，直到接到探测信号。

4. 运动到第二接近点

探针快速 (G00) 从探测点运动到下一个接近点。

该运动分两个阶段执行。

·1· 退回到第一接近点。

·2· 在凸台上运动距离 (B) 到第二接近点。

5. 第二次探测运动

同第一次探测运动。

6. 运动到第三接近点。

与上相同。

7. 第三次探测运动

与上相同。

8. 运动到第四接近点

与上相同。

9. 第四次探测运动

与上相同。

10.退回运动

退回运动分三个阶段 :

·1· 退回到第四接近点。

·2· 在距离凸台上面 (B) 处 ， 探针快速退回到实际凸台中心 （计算的）。

·3· 当编写 (C0), 探针运动到循环调用点。

先沿垂直轴运动到循环调用点在该轴的坐标位置，然后在主工作平面运动到循环调用点。





